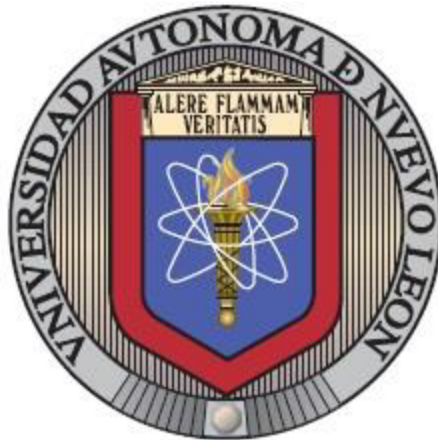


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERIA MECÁNICA Y ELÉCTRICA



**DESARROLLO DE INTERFAZ PARA LECTURA Y ALMACENAMIENTO DE
PARÁMETROS CAN BAJO LA NORMA SAE-J1939.**

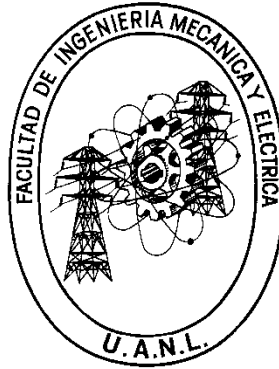
POR

ARMANDO RAMÍREZ CUEVAS

**EN OPCIÓN AL GRADO DE
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA AUTOMOTRIZ**

ABRIL, 2017

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERIA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



**DESARROLLO DE INTERFAZ PARA LECTURA Y ALMACENAMIENTO DE
PARÁMETROS CAN BAJO LA NORMA SAE-J1939.**

POR

ARMANDO RAMÍREZ CUEVAS

**EN OPCIÓN AL GRADO DE
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA AUTOMOTRIZ**

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN,

ABRIL DE 2017

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERIA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO

Los miembros del Comité de Tesis recomendamos que la Tesis Propuesta para el Desarrollo de Interfaz para Lectura y Almacenamiento de Parámetros CAN Bajo la Norma SAE-J1939 realizada por el alumno ingeniero Armando Ramírez Cuevas, con número de matrícula 1757857, sea aceptada para su defensa como opción al grado de Maestría en Ciencias de la Ingeniería Automotriz.

El Comité de Tesis

Dr. Héctor Gilberto Barrón González
Asesor

Dr. Héctor Rafael Siller Carrillo
Revisor

“M.C. Pedro Orta Castañón”
Revisor

Vo. Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, México

a 05 de abril de 2017

DEDICATORIA

Dedico esta investigación a mis padres Armando y Patricia y a mis hermanas Stephanie y Denisse; quienes son parte fundamental en mi vida y en todo momento me proporcionan apoyo moral para seguir adelante y cumplir cada uno de mis propósitos.

AGRADECIMIENTOS

Quiero expresar mis agradecimientos a todos los que estuvieron involucrados durante esta investigación y me apoyaron para llevar a cabo exitosamente este proyecto.

A mi asesor de tesis, el Doctor Héctor Gilberto Barrón González quien en todo momento tuvo la disponibilidad para proporcionarme apoyo técnico y la inteligencia para guiarme por el mejor camino gracias a sus cuestionamientos.

Al Doctor Héctor Rafael Siller Carrillo por darme la oportunidad de ser parte de su equipo de trabajo y poder participar en esta investigación, por su apoyo y disposición a siempre proporcionar las herramientas necesarias para obtener resultados exitosos. Al M.C. Pedro Orta Castañón, por brindarme apoyo científico en todo momento y hacer de esta tesis una mejor.

A la ingeniera Laura Piña y su gran equipo de trabajo, por siempre darme todas las facilidades en planta para realizar pruebas, proporcionar herramientas, equipo y gente capacitada que nutrieran la investigación y resultados de esta tesis.

Por último quiero agradecer el apoyo brindado a todos mis familiares y amigos que estuvieron siempre presentes; principalmente a mis padres Armando y Patricia, quienes han sido pilar importante para llegar a donde estoy y para lograr esta meta. A mis hermanas Stephanie y Denisse quienes fueron motivación y ejemplo para seguir adelante.

Tabla de contenido

DEDICATORIA	4
AGRADECIMIENTOS	5
Capítulo 1. Introducción.	10
1.1 Prefacio	11
1.2 Hipótesis	13
1.3 Objetivos	14
1.4 Estructura de tesis	14
Capítulo 2: Estado del arte	15
2.1 NI: Hardware y software para sistemas de adquisición de datos.	16
2.2 Protocolo SAE J1939	18
2.2.1 Familia de normas de SAE J1939	18
2.2.2 Desarrollos previos de aplicaciones CAN para la industria automotriz pesada con equipo National Instruments.	21
Capítulo 3: Estudio de los requerimientos	23
3.1 Introducción	24
3.2 Metodología	24
3.2.1 Sincronización de aplicación con equipo de National Instruments.	24
3.2.2 Presupuesto.	25
3.2.3 Parámetros recibidos.	25
3.2.4 Equipo robusto y fiable.	25
3.3 Resultados	25
Capítulo 4: Propuesta de interfaz CAN	28
4.1 Introducción	29
4.2 Metodología	29
4.2.1 Identificación de parámetros en Norma SAE J1939-71.	29
4.2.2 Configuración: Sistema de adquisición de datos y Host.	31
4.2.3 Configuración: Interfaz del vehículo y Módulo NI 9862.	33
4.2.4 Desarrollo de código.	40
4.3 Discusiones	44
Capítulo 5: Prueba de concepto	46
5.1 Introducción	47
5.2 Metodología	47

5.2.1 Camión de pruebas.....	47
5.2.2 Preparación previa.....	47
5.2.3 Toma de lecturas.....	49
5.3 Resultados.....	51
5.3.1 Interfaz desarrollada bajo hardware de National Instruments.....	52
5.3.2 Software CANalyzer.....	55
5.3.3. Comparación de resultados.....	61
5.3.4. Porcentaje de error.....	63
5.5 Discusiones.....	65
Capítulo 6: Conclusiones y líneas futuras.....	66
6.1 Conclusión general.....	67
6.2 Conclusiones particulares	67
6.3 Líneas futuras.....	68
Referencias.....	70

Índice de imágenes

Fig. 1. Listado de parámetros.....	12
Fig. 2. NI CRIO-9030.....	16
Fig. 3. Especificaciones CRIO-9030.....	17
Fig. 4. Módulo NI 9862.....	17
Fig. 5. Gráfica Engine Speed.....	21
Fig. 6. Gráfica Fuel Rate.....	21
Fig. 7. Estructura natural de parámetros.....	26
Fig. 8. Estructura corregida de parámetros.....	27
Fig. 9. Estructura parámetro Engine Speed.....	29
Fig. 10. Estructura parámetro Accelerator Pedal Position.....	30
Fig. 11. Estructura parámetro WHeel-Based Vehicle Speed.....	30
Fig. 12. Estructura parámetro Engine Coolant Temperature.....	30
Fig. 13. Conexión Ethernet.....	31
Fig. 14. NI MAX.....	32
Fig. 15. Conexión NI MAX.....	32
Fig. 16. NO conexión NI MAX.....	32
Fig. 17. Módulo en NI MAX.....	33
Fig. 18. Cable J1930 - Serial.....	33
Fig. 19. Pinout SAE J1939/13.....	33
Fig. 20. Pinout módulo NI 9862.....	34
Fig. 21. Conector SAE J1939/13.....	34
Fig. 22. Conector Serial.....	35

Fig. 23. Estructura general de conexión.....	35
Fig. 24. New NI-XNET Session.	36
Fig. 25. Configuración 1 NI-XNET.	37
Fig. 26. Correcta conexión.	38
Fig. 27. Error en conexión.	38
Fig. 28. Configuración 2 NI-XNET.	39
Fig. 29. Sesión creada.....	39
Fig. 30. Block Diagram.....	40
Fig. 31. Front Panel.	40
Fig. 32. Sesión creada.....	41
Fig. 33. Sesión NI-XNET en "Block Diagram".....	41
Fig. 34. Filtrado por PGN.....	41
Fig. 35. Conversión de parámetros.	42
Fig. 36. Panel frontal finalizado.....	43
Fig. 37. Código de almacenamiento.....	44
Fig. 38. Conexión NI MAX.....	48
Fig. 39. Conexión a la red CAN del vehículo.....	49
Fig. 40. Estructura de validación.	50
Fig. 41. Interfaz de usuario.....	50
Fig. 42. Interfaz CANalyzer.	51
Fig. 43. Primer pestaña.	52
Fig. 44. Segunda Pestaña.	53
Fig. 45. Selección "Engine Speed".....	54
Fig. 46. Engine Speed.	54
Fig. 47. Accelerator Pedal Position.	54
Fig. 48. Vehicle Speed.	55
Fig. 49. Engine Coolant Temperature.	55
Fig. 50. CANalyzer exportación de archivo.	56
Fig. 51. Trace Export.....	56
Fig. 52. Guardar archivo.	57
Fig. 53. Archivo .asc de CANalyzer.	58
Fig. 54. Filtrado por PGN.....	59
Fig. 55. Magnitud de Engine Speed.....	60
Fig. 56. Engine Speed – CANalyzer.....	60
Fig. 57. Accelerator Pedal Position – CANalyzer.	61
Fig. 58. Vehicle Speed – CANalyzer.....	61
Fig. 59. Engine Coolant Temperature – CANalyzer.....	61
Fig. 60. Resultados comparativos "Engine Speed".	62
Fig. 61. Resultados comparativos "Accelerator Pedal Position".....	62
Fig. 62. Resultados comparativos "Vehicle Speed".....	63
Fig. 63. Resultados comparativos Engine Coolant Temperature.....	63
Fig. 64. Calculo de porcentaje de error.....	64

Índice de tablas

<i>Tabla 1. Ficha técnica</i>	47
Tabla 2. Parámetros SAE J1939-71.....	58
Tabla 3. Porcentajes de error.....	65

Capítulo 1. Introducción.

A través de este primer capítulo y para enriquecer las bases técnicas del lector se abordarán definiciones clave relacionadas al protocolo CAN J1939, al hardware que se utiliza, al almacenamiento y así mismo se presentará una breve reseña de la industria automotriz de camiones pesados en México. Finalmente se describirá la hipótesis y los objetivos de la investigación de manera específica.

1.1 Prefacio

Hoy en día, México se ha logrado posicionar como un referente importante en la industria automotriz de camiones pesados; la inversión en el desarrollo intelectual, infraestructura y tecnología en esta creciente industria han logrado impulsar un mercado verdaderamente competitivo. De acuerdo a la Asociación Nacional de Productores de Autobuses, Camiones y Tractocamiones A.C. (ANPACT) en México existen 12 plantas manufactureras de vehículos pesados como Kenworth, Freightliner, International, DINA, Cumins, entre otras que juntas en 2015 sumaron una producción total anual de más 150,000 unidades. Hasta este año México se consolida como el séptimo mejor constructor del mundo y como el cuarto mejor exportador de la industria automotriz pesada [1]

Como consecuencia de la competencia y de la gran demanda que existe en esta industria, además de invertir en recursos para construir más vehículos año con año, se busca principalmente aumentar el nivel de confiabilidad, durabilidad, seguridad y de confort en los camiones. Para lograr lo anterior, es fundamental conocer a profundidad el comportamiento del vehículo en diversos escenarios, es decir, tener la capacidad de determinar el nivel de exigencia del motor (consumo de combustible, velocidad del vehículo, revoluciones por minuto, temperatura del anticongelante, entre otros) bajo ambientes determinados para así poder realizar mejoras objetivas.

Internamente y gracias a la tecnología desarrollada hasta hoy en día, un camión, por medio de un lenguaje binario y una red multidireccional, conoce perfectamente cientos de parámetros de diversos sub-sistemas (motor, transmisión, frenos, etc) que le permiten ajustarse de la mejor manera a la exigencia del escenario y del operador para mantenerse siempre dentro de los rangos óptimos de funcionamiento. Por medio de este proyecto, el objetivo es establecer una conexión con esta red interna del vehículo llamada CAN bajo el protocolo SAE J1939.

El protocolo de comunicación SAE J1939 es el que actualmente se utiliza en vehículos pesados, ha sido desarrollado y estandarizado por la “SAE Truck and Bus Control and Communications Network Committee” y fue lanzado a inicios del 2000 como sucesor del protocolo SAE J1708 y SAE J1587; ambos tienen una menor tasa de velocidad (9Kbits/s contra 250Kbits/s del J1939). Los beneficios que ofrece este nuevo protocolo son principalmente la flexibilidad y confiabilidad, lo económico de los módulos, el auto-diagnóstico, la calibración y la lectura de diagnóstico de cada componente individualmente (transmisión, motor, frenos, aire acondicionado, etc), gracias a que todos los sistemas electrónicos se encuentran conectados a una misma red central (ECU) lo que permite por medio de una misma conexión tener acceso a todos los sistemas para efectos de monitoreo y diagnóstico de fallas.

En el mercado existen distintas herramientas que permiten establecer una comunicación en tiempo real y visualizar algunos parámetros que emite la ECU de los vehículos pesados, sin embargo no todas estas herramientas nos ofrece la capacidad de almacenar cada dato adquirido a lo largo de una prueba. Aquí radica la problemática principal, además de un sistema capaz de establecer comunicación con un camión (mediante el protocolo SAE J1939) se requiere que tenga la capacidad de leer, decodificar y almacenar (en unidades ingenieriles) 4 parámetros propuestos por la industria para un análisis posterior.

En la norma expuesta por la SAE J1939-71 se desarrolla puntualmente cientos de parámetros que engloba el protocolo, cada uno con identificadores irrepetibles (SPN, por sus siglas en inglés), con tasas de repetición específicas (TTR) y divididos en grupos de acuerdo a sistemas en el vehículo (PGN). En la siguiente imagen se muestran los 4 parámetros que la empresa exige leer, cada uno de ellos con su respectiva información

<p>Accelerator Pedal Position</p> <p>SPN: 91</p> <p>Data Length: 1 byte</p> <p>Data Range: 0-100%</p> <p>Resolution: 0.4 %per bit</p> <p>Parameter Group Number: 61443 (0x00F003)</p> <p>Transmission Repetition Rate: 50 ms</p> <p>Byte position : 2 of 8</p>	<p>Engine Speed</p> <p>SPN: 190</p> <p>Data Length: 2 bytes</p> <p>Data Range: 0-8,031 RPM</p> <p>Resolution: 0.125 RPM per bit</p> <p>Parameter Group Number: 61444 (0x00F004)</p> <p>Transmission Repetition Rate: 50 ms</p> <p>Byte position : 4-5 of 8</p>
<p>Engine Coolant Temperature</p> <p>SPN: 110</p> <p>Data Length: 1 byte</p> <p>Data Range: -40 to 210 deg C</p> <p>Resolution: 1 deg C per bit</p> <p>Parameter Group Number: 65262 (0x00FEE2)</p> <p>Transmission Repetition Rate: 1 s</p> <p>Byte position : 1 of 8</p>	<p>Wheel-Based Vehicle Speed</p> <p>SPN: 84</p> <p>Data Length: 2 bytes</p> <p>Data Range: 0 to 250.996 km/h</p> <p>Resolution: 1/256 km/h per bit</p> <p>Parameter Group Number: 65265 (0x00FEF1)</p> <p>Transmission Repetition Rate: 100 ms</p> <p>Byte position : 2-3 of 8</p>

Fig. 1. Listado de parámetros.

[2].

Es objetivo de este proyecto de tesis hacer el correcto filtrado para solo leer las magnitudes de estos 4 parámetros que se describieron; estos parámetros deben almacenarse de una manera ordenada, respetando cada tiempo de lectura, en un archivo amigable y de fácil lectura para futuro análisis, a diferencia de lo que ofrecen los sistemas ya desarrollados por diversas marcas como el CANalyzer de Vector que te permite almacenar información con las siguientes restricciones:

- Al ser una herramienta que abarca, además de almacenamiento de parámetros, funciones como lectura de códigos de diagnóstico, calibración de módulos entre

otras cosas es muy complejo de utilizar, requiere de capacitación impartida por el fabricante lo que conlleva tiempo y por consecuencia mayores costos de inversión [3].

- La compatibilidad de las extensiones en que te permite grabar son mayormente para uso exclusivo en softwares que ellos mismo desarrollan o en su defecto, para abrir en softwares no muy amigables y que requiere de compra de licencia, lo que aumenta el costo de inversión [4].
- Son softwares comerciales y patentados que no te permiten modificar o adaptar la aplicación a tu necesidad.

Finalmente se realizó un detallado análisis y una vez que se consideraron las posibilidades de lectura, las de almacenamiento y sobre todo las cuestiones de bajo presupuesto, se decidió por utilizar hardware de National Instruments quien ofrece una amplia variedad de procesadores y respectivos módulos que son compatibles con CAN de alta velocidad y que nos dan la posibilidad de adecuar el desarrollo de aplicación casi a cualquier necesidad con un nivel de complejidad bajo de programación.

De esta manera, para el completo desarrollo de la aplicación, se utilizará lo siguiente: un módulo CAN NI 9862 de alta velocidad [5] y un procesador cRIO 9030 dual core de 1.33 GHz [6] se desarrollará una aplicación que sea capaz de decodificar los 4 parámetros que hasta el momento requiere la empresa.

1.2 Hipótesis

El monitoreo de cualquier sistema siempre es importante para determinar el desempeño del mismo y sobre todo para mantenerlo en óptimas condiciones de rendimiento en diferentes escenarios de aplicación. Utilizando el sistema propuesto en un camión de carga que se moverá en carreteras del país se planea adquirir datos confiables para tomar medidas ingenieriles de mejora, por lo que es de suma importancia que haya coherencia entre la tasa de repetitividad que marca el Manual SAE J1939 y lo que se almacene en el sistema de adquisición de datos; de estos últimos par de puntos se obtienen las siguiente hipótesis:

1. La tasa de repetición (TTR) que se establece en el Manual de la SAE J1939-71 a cada uno de los 4 parámetros a investigar es precisa y no presenta alteraciones.
2. Además de visualizar, es posible almacenar en magnitudes ingenieriles los parámetros recibidos a través la ECU del camión en un archivo TDMS.

1.3 Objetivos

El objetivo general de la presente investigación es el desarrollo de una interfaz versátil, confiable y eficiente que mediante hardware de National Instruments y bajo el protocolo de CAN SAE J1939 gestione la comunicación con la ECU de vehículos pesados para visualizar, decodificar y almacenar 4 parámetros solicitados por la industria, y el cual deberá cumplir con los siguientes requerimientos:

1. Interfaz amigable, confiable, robusta y que se enlace a la red CAN de vehículos pesados bajo la Norma SAE-J1939.
2. Adecuación de la interfaz otros sistema de adquisición de datos desarrollado con hardware de National Instruments.
3. Generar archivo TDMS en cuya estructura se almacene dividido por columnas los 4 parámetros que indique la magnitud de cada uno de ellos además de una quinta columna que indique el tiempo de lectura de cada parámetro.

1.4 Estructura de tesis

La presente tesis está organizada de la siguiente manera: Capítulo 2, estado del arte, se profundiza acerca de sistemas de adquisición de datos utilizados en vehículos pesados más proyectos acerca del protocolo CAN J1939. Capítulo 3, estudio de los requerimientos de hardware para el desarrollo de la interfaz. Capítulo 4, la propuesta de interfaz CAN para vehículos que utilicen la norma SAE-J1939. Capítulo 5, prueba de concepto en campo y finalmente capítulo 6, conclusiones y líneas futuras de la interfaz.

Capítulo 2: Estado del arte

Este capítulo presenta el estado del arte relacionado a la investigación, y se encuentra dividido en dos secciones: La primera en sistemas de adquisición de datos, aplicado preferentemente a la industria automotriz pesada. La segunda enfocada al protocolo de comunicación CAN J1939 de la SAE International. En ambas secciones se profundizará sobre lo que se ha investigado para determinar un punto de partida y el impacto científico que se generará al finalizar este presente proyecto.

2.1 NI: Hardware y software para sistemas de adquisición de datos.

National Instruments (NI) es una empresa fundada en Estados Unidos en el año de 1976 [7] siendo en la actualidad una empresa líder en el desarrollo de hardware y software modular, y que además se caracteriza por tener un desarrollo de sistemas en programación gráfica. Uno de los softwares de mayor éxito que han desarrollado y que ha estado en el mercado académico e industrial desde la década de los 90 es LabVIEW. Al utilizar un entorno de desarrollo de programación gráfica, Labview facilita visualizar, crear y codificar sistemas de ingeniería con un ahorro considerable de tiempo. [8].

LabVIEW cuenta con diversos bloques o módulos de programación que te permiten ampliar las herramientas a utilizar y por ende a hacer códigos más impactantes, como FPGA, NI-XNET, Real Time, Veristand, entre otros, para uso efectivo de esta aplicación se requiere cuando menos contar con NI-XNET [9] y Real Time [10], en el capítulo 4 se mencionan puntualmente las razones. Otro software que desarrolla National Instruments y que es indispensable para el presente desarrollo de la aplicación es NI MAX

Entre el diverso mundo de Hardware que desarrolla National Instruments se encuentra una línea robusta de procesadores llamada CompactRIO; Esta línea consiste en sistemas avanzados de procesamiento de datos altamente integrados que tienen la versatilidad de trabajar con hasta 8 módulos de entradas/salidas analógicas y digitales, temporizadores/contadores, comunicación CAN y LIN que en conjunto a su lenguaje de programación gráfico hacen de este dispositivo una fuerte, confiable, certificada y accesible (económicamente) herramienta [11].

Para efectos de esta presente investigación, luego de hacer una importante comparación entre los diversos cRIO's que ofrece National Instruments en cuanto a precio, capacidad de procesamiento, dimensiones y cantidad de ranuras para módulos, se optó por el CompactRIO-9030 (Fig. 2). A continuación se muestra una tabla con sus especificaciones más importantes [6].



Fig. 2. NI CRIO-9030.

General

1.33GHz Dual Core Intel Atom Processor E3825.

NI Linux Real-Time con UI embebida.

Memoria DDR3 de 1 GB.

Almacenamiento no volátil de 4 GB.

Tecnología Real-Time y FPGA (Kintex-7).

Temperatura de operación de -20 a 55 °C.

Alimentación de 9 a 30 V.

Puerto Periférico

2 Gigabit Ethernet.

1 Serial RS232.

1 Serial S485.

2 USB Hi-Speed.

1 USB de dispositivo (MiniDisplay).

Ranura para tarjeta SDHC.

4 ranuras para módulos NI serie C.

Fig. 3. Especificaciones CRIO-9030.

En conjunto con el CompactRIO anteriormente descrito se utiliza un módulo de la serie C del mismo fabricante: NI 9862 (Fig. 4) que cuenta con un puerto de alta velocidad (hasta 8 Mb/s) y la interfaz para trabajar bajo protocolos CAN. Además cuenta con NI-XNET, aplicación que permite desarrollar bases de datos, ya sea para visualizar e integrarte a una red CAN ya existente (como es nuestro caso) o para desarrollar una propia red CAN y sus respectivas tramas [5].



Fig. 4. Módulo NI 9862.

Para mayores referencias, durante la parte final del apartado 2.2 se describirá un proyecto de investigación desarrollado con el equipo de National Instruments antes descrito.

2.2 Protocolo SAE J1939

Desde su lanzamiento en Abril del 2000 el protocolo SAE J1939 ha tenido una gran aceptación como protocolo de comunicación de vehículos pesados. Este protocolo está conformado por una red de comunicación de alta velocidad (250 ks/s), ligera y multidireccional con la finalidad de proveer una interconexión abierta entre los diversos sistemas electrónicos del vehículo.

2.2.1 Familia de normas de SAE J1939

Para efectos de practicidad el protocolo de la SAE J1939 está dividido en distintos documentos, en cada uno aborda de manera puntual distintas sub-áreas en dónde y cómo se aplica este protocolo, a continuación se describe brevemente cada uno:

J1939-01

Red de control y comunicación de equipos en carretera (On-highway Equipment Control and Communications Network).

Esta norma contiene los requerimientos necesarios para camiones y trenes que circularán en carretera, se entiende por estos últimos por vehículos que remolcan cualquier otro tipo de vehículo, por ejemplo un tráiler con una caja independiente [12].

J1939-02

Red de control y comunicación de máquinas agrícolas y forestales (Agricultural and Forestry Off-Road Machinery Control and Communications Network).

Como su nombre lo indica, el objetivo de este documento es especificar los requerimientos de comunicación del protocolo SAE J1939 para equipos de construcción y agricultura, por ejemplo tractores, plantas generadoras de energía, excavadoras, etc. [13].

J1939-03

Guía de implementación de diagnóstico a bordo (On-Board Diagnostics Implementation Guide).

Este documento especifica la implementación de diagnóstico a bordo, mejor conocido como OBD (por sus siglas en inglés) y detalla la interoperabilidad entre computadoras y diversas herramientas de diagnóstico de fallas y vehículos pesados [14].

J1939-05

Guía de la implementación de la unidad marina de impulsión de popa y encendido por chispa (Marine Stern Drive and Inboard Spark-Ignition Engine On-Board Diagnostics Implementation Guide).

Describe la aplicación del protocolo SAE J1939 en unidades marinas y regula el sistema de diagnóstico de detección de fallas de estas unidades [14].

J1939-11

Capa física – 250k bits/s, par trenzado blindado (Physical Layer – 250k bits/s, Twisted Shielded Pair).

Esta norma define el número total de sistemas electrónicos que se pueden conectar físicamente al bus de la red CAN, esto se verá limitado por la cantidad y rapidez de las cargas eléctricas que sean emitidas dentro de la red [15].

J1939-13

Conector de diagnóstico fuera de sistema (Off-Board Diagnostics Connector).

Este documento describe el conector externo (OBD) para diagnóstico con el que cuenta el vehículo para obtener acceso a los enlaces de comunicación del mismo [16].

J1939-15

Capa física reducida – 250k bits/s, par trenzado no blindado (Reduced Physical Layer, 250K bits/sec, Un-shielded Twisted Pair (UTP)).

Esta norma define los mismos requerimientos para la conexión de sistemas que el documento J1939-11 pero utilizando un par de cables trenzados sin blindaje/cubierta [17].

J1939-21

Capa de enlace de datos (Data Link Layer).

Este documento describe la capa de enlace de datos y se establece que dentro del protocolo SAE J1939 no hay otra alternativa permitida más que tramas compuestas por 29-bits de identificación [18].

J1939-31

Capa de red (Network Layer).

Establece que aplica únicamente a las unidades de control del motor (ECU's) que están diseñadas para establecer servicios de red, esto sería cuando una ECU de un vehículo debe mantener una comunicación con otra ECU de un vehículo distinto [19].

J1939-71

Capa de aplicación del vehículo (Vehicle Application Layer).

Es el manual más extenso de este protocolo, en este se describe la estructura de cómo está conformada cada trama y cada cuanto tiempo la envía determinado sistema

electrónico; para esta investigación se trabajó exclusivamente con este documento que apoyó en la construcción de una base de datos que nos ayudó a interpretar en unidades ingenieriles los parámetros recibidos en sistema binario [20].

J1939-73

Capa de aplicación – Diagnósticos (Application Layer – Diagnostics).

En este documento se profundiza acerca de los códigos establecidos para los mensajes de servicio de diagnóstico; estos mensajes de diagnóstico proporcionan orientación precisa de determinadas fallas para que el vehículo sea reparado [21].

J1939-74

Aplicación – Mensajes configurables (Application – Configurable Messaging).

Aquí se describe la estructura de mensajes para un conjunto de mensajes, lo cual permite al usuario determinar y anunciar a otros en la red [22].

J1939-75

Capa de aplicación – Generadores industriales (Application Layer - Generator Sets and Industrial).

En esta norma se define la estructura de los parámetros asociados al monitoreo y control de generadores y equipos relacionados a la energía eléctrica [23].

J1939-81

Administración de redes (Network Management).

La función principal de esta norma es la de detección y notificación de errores relacionados con la red, ajenos al comportamiento de los sistemas que componen la red CAN [24].

J1939-82

Cumplimiento – Camión y autobús (Compliance – Truck and Bus).

Especifica los procedimientos necesarios para que determinado fabricante de ECU's genere sus propios documentos de prueba y así asegurar el correcto funcionamiento dentro de la red SAE J1939 [25].

J1939-84

Cajas de prueba de cumplimiento de comunicaciones OBD para componentes y vehículos pesados (OBD Communications Compliance Test Cases for Heavy Duty Components and Vehicles)

Este manual está conformado por ensayos, mensajes de pruebas de diagnóstico, metodologías y resultados con la finalidad de verificar y asegurar que los vehículos y/o componentes que salgan al mercado sean capaces de comunicarse correctamente [26].

2.2.2 Desarrollos previos de aplicaciones CAN para la industria automotriz pesada con equipo National Instruments.

Recientemente, la Universidad de Nebraska Lincoln en USA publicó una investigación titulada “Validation of machine CAN bus J1939 fuel rate accuracy using Nebraska Tractor Test Laboratory fuel rate data” en donde debían adquirir datos de un tractor que funciona bajo el protocolo SAE J1939 con el objetivo principal de hacer una comparación entre dos parámetros adquiridos desde la ECU principal del vehículo y lo medido físicamente por sensores externos al vehículo. Los dos parámetros a comparar fueron: Engine Speed (PGN F004, SPN 190 a 100 Hz) y Engine Fuel Rate (PGN FEF2, SPN 183 a 10 Hz).

Para cumplir el objetivo principal de su investigación instrumentaron el tractor con un sensor digital de fibra óptica para leer las revoluciones del motor y con un sensor de flujo de masas de micro movimiento para determinar la tasa de consumo de combustible. Las pruebas consistieron en lo siguiente: durante un tiempo específico y con el vehículo encendido pero detenido, se almacenaron simultáneamente los parámetros de las dos formas antes descritas, una por medio de la ECU principal del vehículo y su red CAN y la segunda por medio de los sensores ajenos al vehículo, a continuación se muestran un par de gráficas presentadas en sus resultados (Fig. 5 y 6) [27].

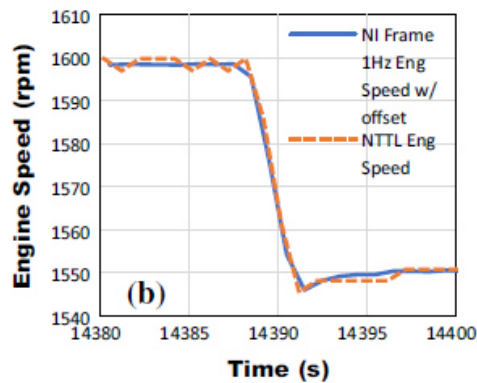


Fig. 5. Gráfica Engine Speed.

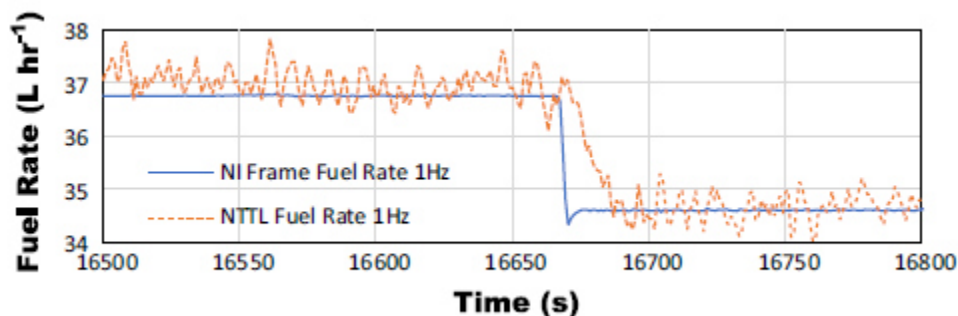


Fig. 6. Gráfica Fuel Rate.

Como se puede observar en la primera gráfica, el resultado final de una prueba es bastante similar en ambas comparaciones a diferencia de la segunda que aunque presenta un comportamiento similar del dominio, muestra picos considerables en comparación a la señal recibida de la ECU.

A pesar que el objetivo principal y el enfoque que toman durante su investigación no impacta directamente a nuestros objetivos, es de suma importancia e interés porque hay ciertas etapas que indiscutiblemente se tienen en común, principalmente el hardware y software que utilizaron para su investigación: un módulo serie C, NI 9862 de un puerto para CAN (mismo que para este presente proyecto) y un equipo cDAQ (chasis que cuenta exclusivamente con diversas ranuras para insertar módulos y no con un procesador integrado) también de National Instruments, tiene como principal diferencia al CRIO 9030 que se ha utilizado para esta tesis que no cuenta con procesador para la adquisición de datos, es decir que todo lo procesa por medio de la computadora a la que está conectado y con la que se programó mientras que el CRIO 9030 puede trabajar perfectamente sin requerir de una computadora externa.

Capítulo 3: Estudio de los requerimientos

Durante este tercer capítulo se abordarán temas referentes a la metodología aplicada para cumplir satisfactoriamente con el proyecto. Se darán a conocer los requerimientos mínimos que solicitó la empresa asociada, se mencionará cómo será implementada esta aplicación en la industria y como parte final, de qué manera se validará su correcto funcionamiento.

3.1 Introducción

A continuación se describirá la metodología que se siguió para hacer la correcta elección del hardware de acuerdo a los requerimientos, también aquí desarrollados, que estableció la industria.

3.2 Metodología

Debido a que el desarrollo del presente proyecto nace por una necesidad real en la industria automotriz pesada, previo a iniciar con cualquier tipo de investigación y propuesta, es necesario conocer a detalle los requerimientos que la empresa exige para cubrir sus expectativas mínimas, los cuales se detallan a continuación.

3.2.1 Sincronización de aplicación con equipo de National Instruments.

En paralelo al desarrollo de la presente aplicación, la empresa, con tecnología de National Instruments, desarrolla otro sistema de adquisición de datos con determinadas características de lectura que complementan la lectura adquirida de parámetros CAN; el requerimiento inicial radica en que la aplicación CAN sea capaz de sincronizarse al otro sistema DAQ, principalmente en el archivo final en donde se muestra la recolección total de los datos adquiridos.

Simultáneamente, se evaluó la opción de adquirir un equipo de lectura de datos CAN que se encuentran a la venta y la de desarrollar el propio sistema a medida. En el mercado existen equipos, como el que se describió en capítulos anteriores, CANalyzer, que te permiten visualizar y almacenar parámetros recibidos por la computadora central del vehículo, sin embargo presentan las siguientes restricciones:

- Son desarrollados con códigos “cerrados” y que no es posible modificar.
- Son mucho más complejos.
- Manejan precios elevados.

El desarrollo de la aplicación CAN como tal, no exige la elección de determinada marca de hardware, pero sí la sincronización de esta nueva aplicación a otro sistema de adquisición de datos ya utilizado por la empresa y desarrollado con equipo de NI; para mayor practicidad y con la finalidad de tener la posibilidad de utilizar ambos equipos de manera sincrónica en un mismo procesador y sobre todo en el condensado de los resultados, se recomendó investigar qué hardware ofrece NI al mercado relacionado con tecnología de comunicación CAN.

3.2.2 Presupuesto.

Por cuestiones de financiamiento académicas y para que fuera atractivo para la industria en donde se aplicará, el equipo a desarrollar tenía restricciones en costo de inversión muy bajos comparados con equipos en el mercado con capacidades similares. Este es uno de los principales requerimientos que limitan las posibilidades de selección de hardware.

3.2.3 Parámetros recibidos.

En temas de lectura, almacenamiento y visualización de los parámetros recibidos por medio de la red CAN, es necesario mantener una interfaz amigable para el usuario en donde pueda observar en tiempo real las lecturas adquiridas por medio de la ECU principal, mientras que automáticamente se deben almacenar en un archivo TDMS con todos los datos para posterior post procesamiento de los mismos.

A continuación se enlistan los parámetros que la empresa solicita visualizar y almacenar:

- Time Data Recorder (Tiempo de lectura).
- Engine Speed (Revoluciones del motor).
- Accelerator Pedal Position (Posición del pedal de aceleración).
- Vehicle Speed (Velocidad del vehículo).
- Engine Coolant Temperature (Temperatura del refrigerante del motor).

Finalmente y después de una corrida de determinado tiempo se creará el archivo mencionado.

3.2.4 Equipo robusto y fiable.

Como se ha comentado, debido a que la aplicación del sistema final estará sujeta a uso en campo y sobre un camión en ruta en alguna carretera del país, es necesario desarrollar un sistema robusto, que resista vibraciones excesivas bajo ambientes de temperatura altos con un alto nivel de exigencia en procesamiento de señales y sobre todo que no descuidara la fiabilidad de los datos recolectados.

3.3 Resultados

Los resultados esperados por la industria se resumen en la figura XX, en donde se puede observar las 4 columnas de parámetros adquiridos más el respectivo tiempo de lectura.

Como especifica el protocolo, los parámetros de Engine Speed y Accelerator Pedal Position tienen una tasa de repetición de 50 ms, Vehicle Speed de 100 ms y por último Engine Coolant Temperature de 1000 ms [20] por lo que es evidente concluir que el almacenamiento, de manera natural, no podrá mantener una coherente relación en lectura por lectura entre los 4 parámetros, como se muestra en la figura 7.

	Time (s)	Engine Speed (RPM)	Accelerator Pedal Position (%)	Vehicle Speed (Km/hr)	Engine Coolant Temperature (Celcius Deg)
1	●	●	●	●	●
2	●	●	●	—	—
3	●	●	●	●	—
4	●	●	●	—	—
5	●	●	●	●	—
6	●	●	●	—	—
7	●	●	●	●	—
8	●	●	●	—	—
9	●	●	●	●	—
10	●	●	●	—	—
11	●	●	●	●	—
12	●	●	●	—	—
13	●	●	●	●	—
14	●	●	●	—	—
15	●	●	●	●	—
16	●	●	●	—	—
17	●	●	●	●	—
18	●	●	●	—	—
19	●	●	●	●	—
20	●	●	●	—	—
21	●	●	●	●	●

Fig. 7. Estructura natural de parámetros.

Sin embargo, gracias a la versatilidad que nos da desarrollar nuestro propio código de programación, esto se corrige repitiendo el último dato obtenido por cada parámetro hasta que no se tenga uno nuevo; tomando como referencia la tasa de repetición más alta de los dos parámetros y con apoyo en la figura 7 se ejemplifica que hay que repetir 20 veces el mismo dato del parámetro Engine Coolant Temperature para tener otro distinto, así como repetir 2 veces la velocidad del vehículo (Vehicle Speed) antes de tener un dato diferente.

De esta manera, corrigiendo el problema de repetitividad, se espera encontrar un archivo que para cualquier tiempo determinado cuente con dato en los 4 parámetros y omitir lo huecos que de manera natural se crearían, para mayor referencia se representa en la figura 8.

	Time (s)	Engine Speed (RPM)	Accelerator Pedal Position (%)	Vehicle Speed (Km/hr)	Engine Coolant Temperature (Celcius Deg)
1	T1	RPM1	%1	Kh/hr1	Celcius1
2	T2	RPM2	%2	Kh/hr1	Celcius1
3	T3	RPM3	%3	Kh/hr2	Celcius1
4	T4	RPM4	%4	Kh/hr2	Celcius1
5	T5	RPM5	%5	Kh/hr3	Celcius1
6	T6	RPM6	%6	Kh/hr3	Celcius1
7	T7	RPM7	%7	Kh/hr4	Celcius1
8	T8	RPM8	%8	Kh/hr4	Celcius1
9	T9	RPM9	%9	Kh/hr5	Celcius1
10	T10	RPM10	%10	Kh/hr5	Celcius1
11	T11	RPM11	%11	Kh/hr6	Celcius1
12	T12	RPM12	%12	Kh/hr6	Celcius1
13	T13	RPM13	%13	Kh/hr7	Celcius1
14	T14	RPM14	%14	Kh/hr7	Celcius1
15	T15	RPM15	%15	Kh/hr8	Celcius1
16	T16	RPM16	%16	Kh/hr8	Celcius1
17	T17	RPM17	%17	Kh/hr9	Celcius1
18	T18	RPM18	%18	Kh/hr9	Celcius1
19	T19	RPM19	%19	Kh/hr10	Celcius1
20	T20	RPM20	%20	Kh/hr10	Celcius1
21	T21	RPM21	%21	Kh/hr1	Celcius2

Fig. 8. Estructura corregida de parámetros.

Capítulo 4: Propuesta de interfaz CAN

En este cuarto capítulo se describirá detalladamente la metodología que se siguió para el desarrollo de la aplicación CAN, desde la configuración de equipo hasta el código de programación para finalizar con la sección de conclusiones, en donde se abordarán las decisiones que aportaron resultados exitosos, las problemáticas encontradas a lo largo del proyecto, así como las posibles mejoras que presenta la aplicación.

4.1 Introducción

En esta sección se demostrará que es sumamente crítico el profundo conocimiento del protocolo de comunicación Controller Area Network y principalmente de la norma SAE J1939-71 quien es la que estipula la estructura de todos los parámetros que funcionan en un vehículo pesado.

4.2 Metodología

Para desarrollar la propuesta de la interfaz CAN bajo la norma SAE-J1939 es necesario identificar los parámetros de lectura dentro de esta norma, posteriormente realizar configuraciones de equipo para finalmente desarrollar el código de lectura y almacenamiento de datos.

4.2.1 Identificación de parámetros en Norma SAE J1939-71.

El primer punto es identificar los parámetros requeridos en la norma oficial de la SAE J1939-71, así encontraremos la trama a la que pertenece cada parámetro y la manera de decodificarlo. En las siguientes imágenes se muestran los 4 parámetros con su respectiva información.

Engine Speed	
Actual engine speed which is calculated over a minimum crankshaft angle of 720 degrees divided by the number of cylinders.	
SPN:	190
Data Length:	2 bytes
Data Range:	0-8,031 RPM
Resolution:	0.125 RPM per bit
Parameter Group Number:	61444 (0x00F004)
Transmission Repetition Rate:	50 ms
Byte position :	4-5 of 8

Fig. 9. Estructura parámetro Engine Speed.

Accelerator Pedal Position

For on-highway vehicles, this will typically be the operator's accelerator pedal. Although it is used as an input to determine powertrain demand, it also provides anticipatory information to transmission and ASR algorithms about driver actions.

SPN:	91
Data Length:	1 byte
Data Range:	0-100%
Resolution:	0.4 %per bit
Parameter Group Number:	61443 (0x00F003)
Transmission Repetition Rate:	50 ms
Byte position :	2 of 8

Fig. 10. Estructura parámetro Accelerator Pedal Position.

Wheel-Based Vehicle Speed

Speed of the vehicle as calculated from wheel or tailshaft speed.

SPN:	84
Data Length:	2 bytes
Data Range:	0 to 250.996 km/h
Resolution:	1/256 km/h per bit
Parameter Group Number:	65265 (0x00FEF1)
Transmission Repetition Rate:	100 ms
Byte position :	2-3 of 8

Fig. 11. Estructura parámetro Wheel-Based Vehicle Speed.

Engine Coolant Temperature

Temperature of liquid found in engine cooling system.

SPN:	110
Data Length:	1 byte
Data Range:	-40 to 210 deg C
Resolution:	1 deg C per bit
Parameter Group Number:	65262 (0x00FEEE)
Transmission Repetition Rate:	1 s
Byte position :	1 of 8

Fig. 12. Estructura parámetro Engine Coolant Temperature.

Es de vital importancia conocer la longitud y posición del parámetro en la trama, de otra manera, al conectarnos a la red CAN del camión solo recibiríamos una enorme cantidad de tramas sin saber lo que significa cada una de ellas, en cambio, teniendo la “familia” a la que pertenece (PGN) es sencillo hacer el primer filtrado de acuerdo a

PGN's, para posteriormente filtrar por posición del byte y finalmente hacer las ecuaciones necesarias para obtener el resultado directamente en la magnitud correspondiente al SPN, en puntos más adelante se explica detalladamente esta última parte.

4.2.2 Configuración: Sistema de adquisición de datos y Host.

Primeramente se realiza la conexión de alimentación del procesador de datos CRIO 9030 a una fuente de 9-32 VDC. Ya alimentado, por medio de un cable Ethernet se conecta el hardware de adquisición con el Host (Computadora en donde se desarrollará la aplicación), el procesador de National Instruments con el que se cuenta tiene dos puertos Ethernet disponibles, es indiferente en qué puerto se realice la conexión. A continuación, en la figura 13 se muestra un ejemplo de la alimentación y la conexión Ethernet.

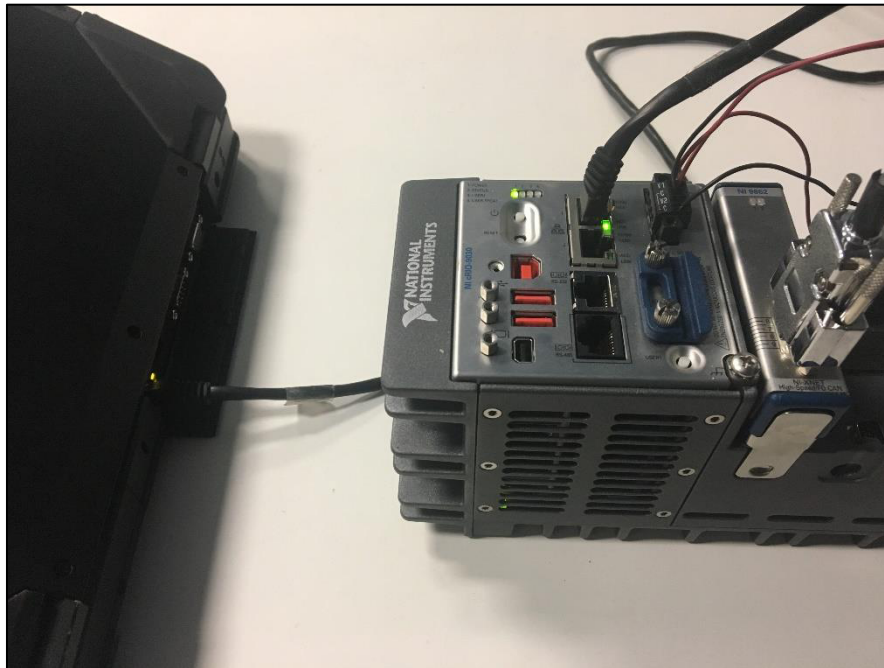


Fig. 13. Conexión Ethernet.

Después de realizar una conexión a una fuente de alimentación de 9-36 VDC, el siguiente paso es abrir el software de National Instruments, Measurements and Automation Explorer (NI MAX). En la pantalla de inicio se presiona la sección “Remote Systems” (Fig. 14), al transcurrir algunos segundos y de manera automática bajo la opción “Remote Systems” se deberá detectar el equipo con la palabra “connected”, en su costado izquierdo, para mayor referencia, se mostrará un ícono gris que representa el sistema de adquisición de datos, esto indica la conexión exitosa (Fig. 15), en caso contrario un ícono en color rojo (Fig. 16).

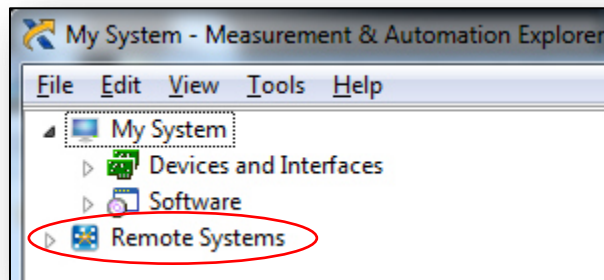


Fig. 14. NI MAX.

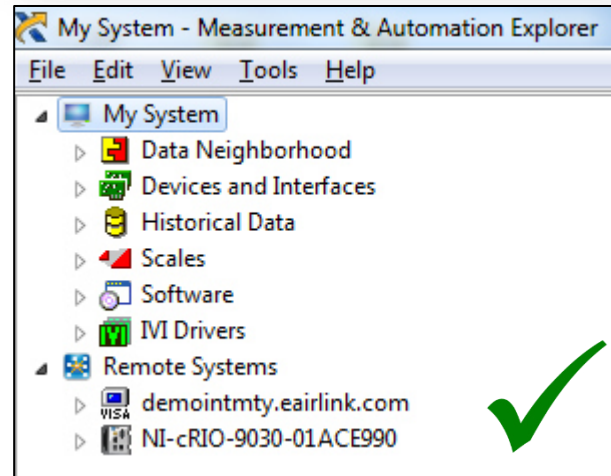


Fig. 15. Conexión NI MAX.

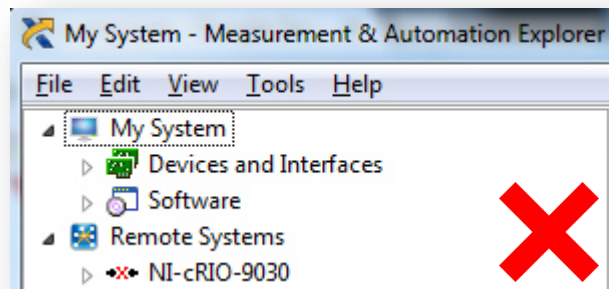


Fig. 16. NO conexión NI MAX.

Finalmente, al expandir el árbol del hardware se mostrarán dos secciones más: “Devices and Interfaces” y “Software”, se expande la sección “Devices and Interfaces” y se deberá mostrar el módulo NI 9862 (Fig. 17), esto indica que el módulo ha sido detectado con éxito por lo que se puede proceder a configurar la conexión Vehículo – Sistema de adquisición de datos.

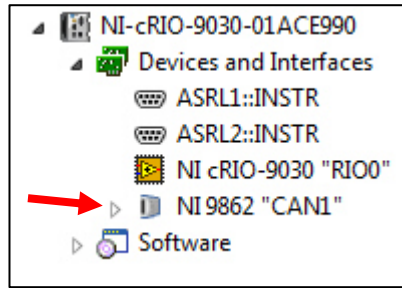


Fig. 17. Módulo en NI MAX.

4.2.3 Configuración: Interfaz del vehículo y Módulo NI 9862.

Habiéndose establecido la conexión correcta entre el procesador y módulo con el Host principal, se procede a establecer conexión entre la red CAN del vehículo y el módulo NI 9862 a través de un cable que se realizó específicamente para este presente proyecto (Fig. 18).



Fig. 18. Cable J1930 - Serial.

Con apoyo en el manual de SAE J1939-13 (Off-Board Diagnostic Connector) se obtuvo el diagrama de los pines del conector de la red CAN que tiene el vehículo (Fig. 19) y con apoyo en el manual de operación del módulo NI 9862, de la misma manera se obtuvo la descripción de los pines del puerto serial que tiene este módulo (Fig. 20).

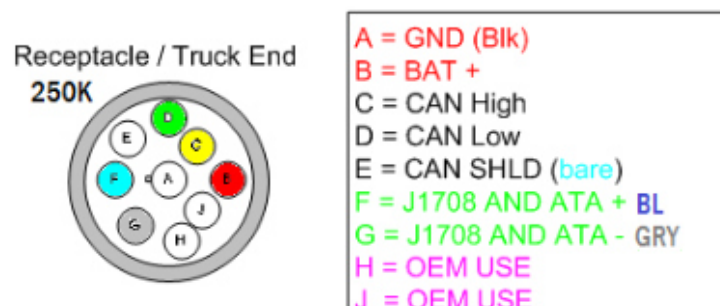


Fig. 19. Pinout SAE J1939/13.

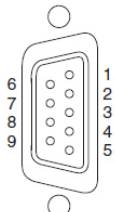
Connector	Pin	Signal
	1	No Connection (NC)
	2	CAN_L
	3	COM
	4	NC
	5	SHLD
	6	COM
	7	CAN_H
	8	NC
	9	V _{SUP}

Fig. 20. Pinout módulo NI 9862.

De esta manera se logró fabricar el cable para establecer la conexión correspondiente entre ambos sistemas, en un extremo el conector de 9 pines J1939 (Fig. 21) y en el otro un puerto Serial de también 9 pines (Fig. 22).



Fig. 21. Conector SAE J1939-13.



Fig. 22. Conector Serial.

El cable J1939-Serial se conecta en un extremo al vehículo y en el otro al módulo 9862 de National Instruments, para mayor referencia y apoyado en la siguiente imagen, hasta el momento se tiene la siguiente estructura de conexión.

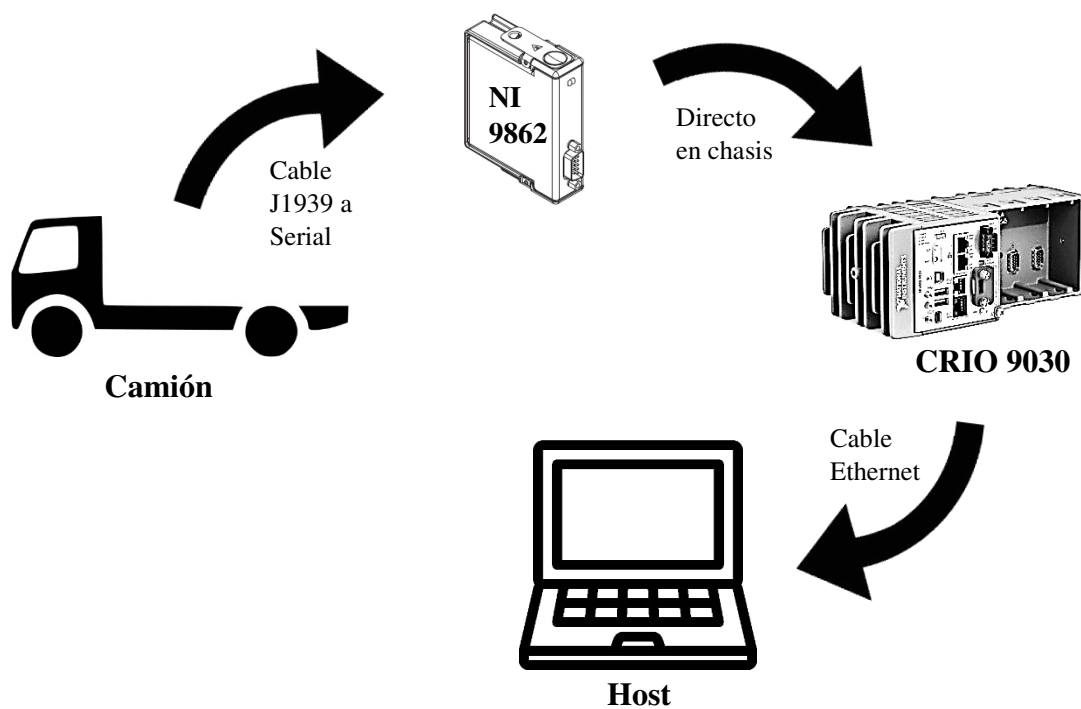


Fig. 23. Estructura general de conexión.

Para finalizar este apartado, se necesita crear una sesión virtual en LabVIEW de acuerdo a la interfaz con la que se va a interactuar, una configuración del módulo NI 9862 que establezca la velocidad de la red a la que se conectará (250 kbits/s en este caso), si se va a utilizar para leer o generar señales o si se va a utilizar para leer o generar tramas completas y qué estructura tiene cada una y por último el protocolo de comunicación.

Con apoyo en la figura 24 se puede observar de qué manera desde LabVIEW se genera esta nueva sesión dando clic derecho en “My Computer”, clic en la sección “New” y finalmente clic en “NI-XNET Session”.

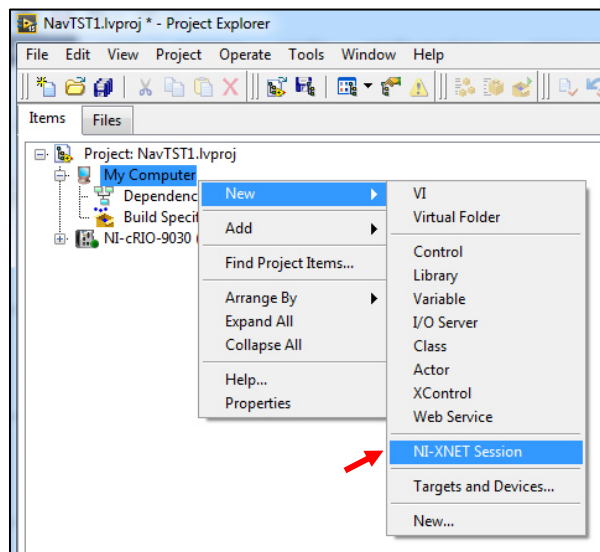


Fig. 24. New NI-XNET Session.

En la ventana emergente, como la que aparece en la Fig. 25, se mostrará la sección que permitirá configuraciones entre señales simples y tramas tanto de entrada como de salida. Como para este proyecto se enlazará a una red CAN ya existente que se comunica a través de diversas tramas (PGN's) a una velocidad específica seleccionamos la opción “Frame Input” y “Stream”, para finalizar esta parte se oprime el botón “Finish”.

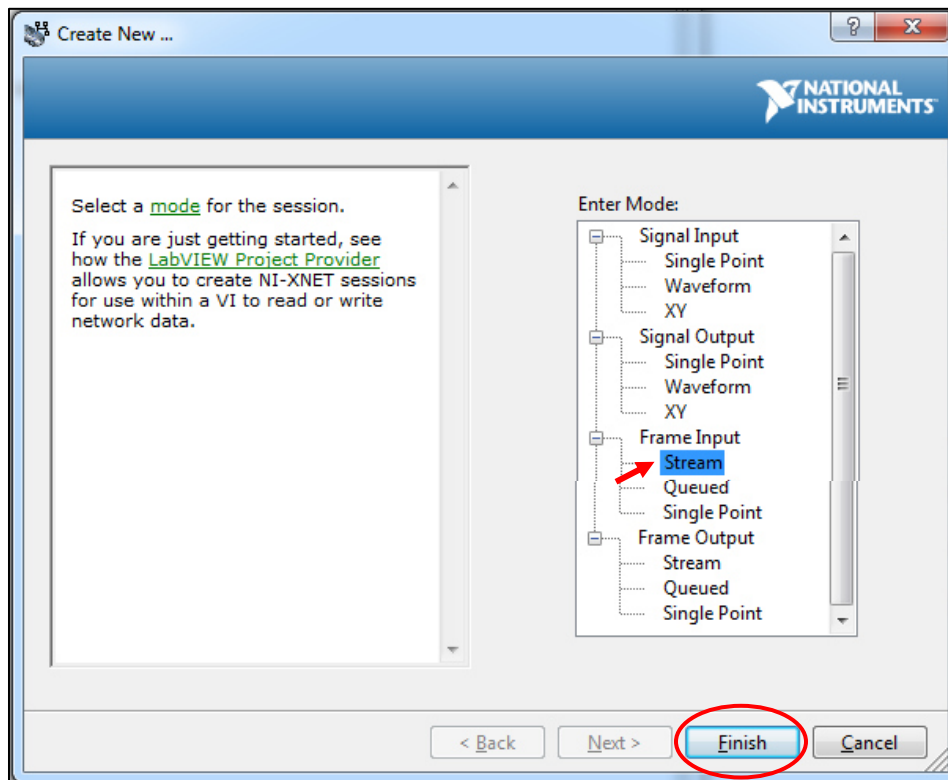


Fig. 25. Configuración 1 NI-XNET.

Si se mantiene una conexión correcta, en la ventana que aparecerá a continuación, se presentarán dos pestañas: “Hardware Selection” y “Cluster Selection”, en la primera pestaña se encontrará una sección que permite asignarle nombre a la sesión, y debajo de ella una sección con el nombre “Interface” que muestra el puerto habilitado, en este caso seleccionamos “Port 1” (Fig. 26). En caso que no exista ningún puerto para seleccionar (Fig. 27) es necesario revisar nuevamente la conexión en el conector del vehículo (J1939-13) y el puerto serial del módulo NI 9862.

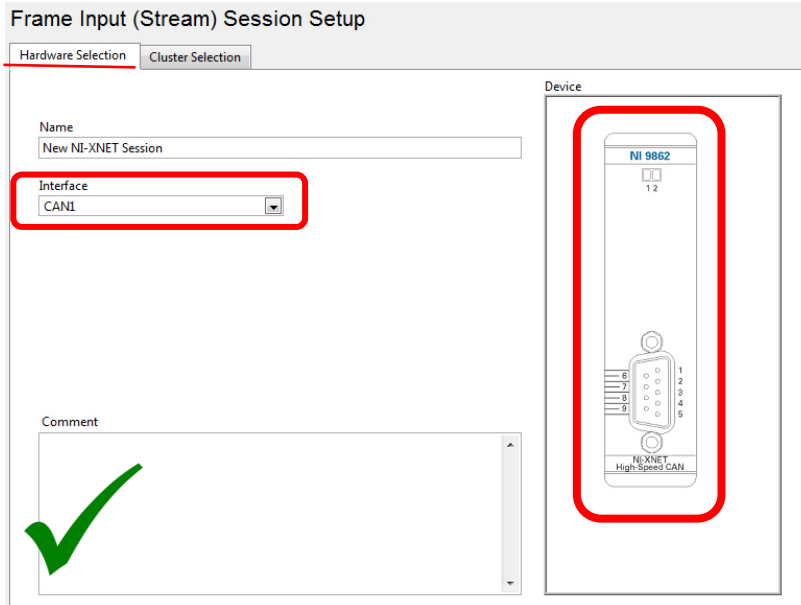


Fig. 26. Correcta conexión.

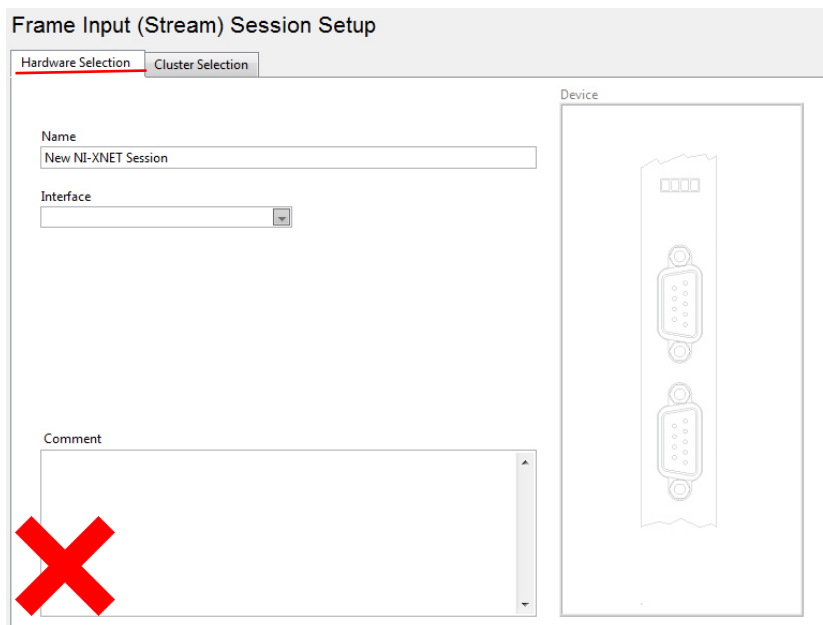


Fig. 27. Error en conexión.

En la misma ventana se selecciona la pestaña "Cluster Selection", también se encontrarán dos secciones: "Database" y "Cluster"; en la sección de "Database" se elige la opción "NIXNET_example" porque es la que permitirá seleccionar el protocolo de comunicación J1939. En la siguiente sección de "Cluster" se selecciona "J1939_Over_CAN" (Fig. 28) y para finalizar se presiona el botón "OK".

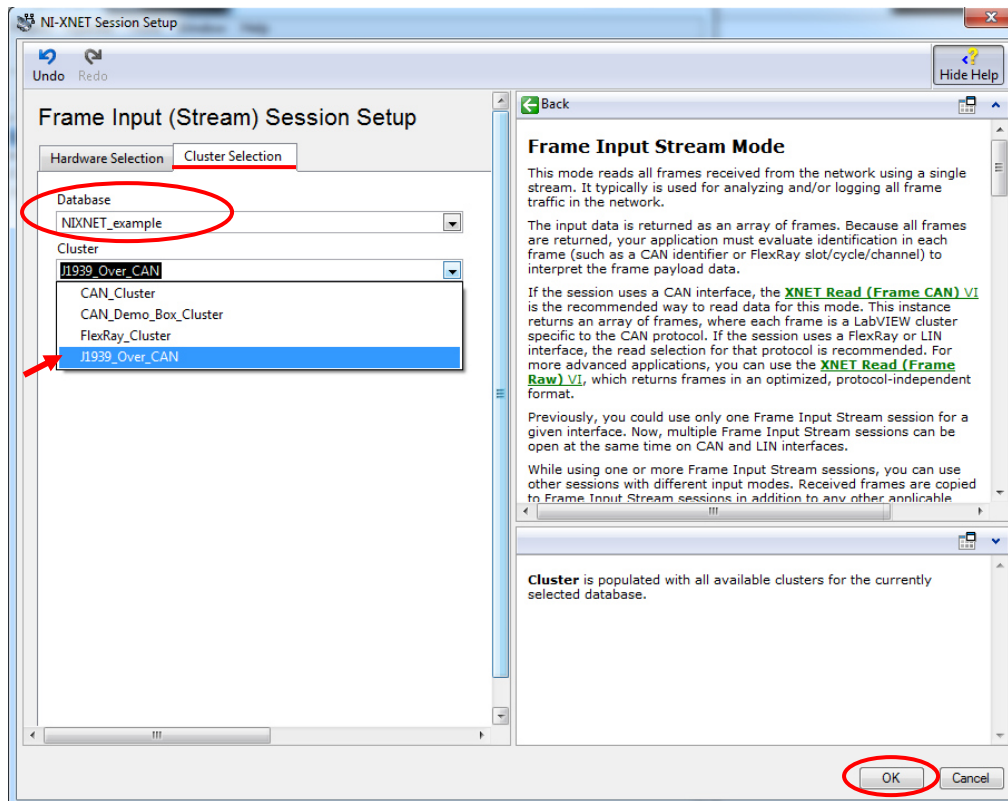


Fig. 28. Configuración 2 NI-XNET.

Para finalizar esta apartado se debe asegurar que aparezca habilitada la nueva sesión que se acaba de generar debajo de “My Computer” en la ventana principal de LabVIEW (Fig. 29).

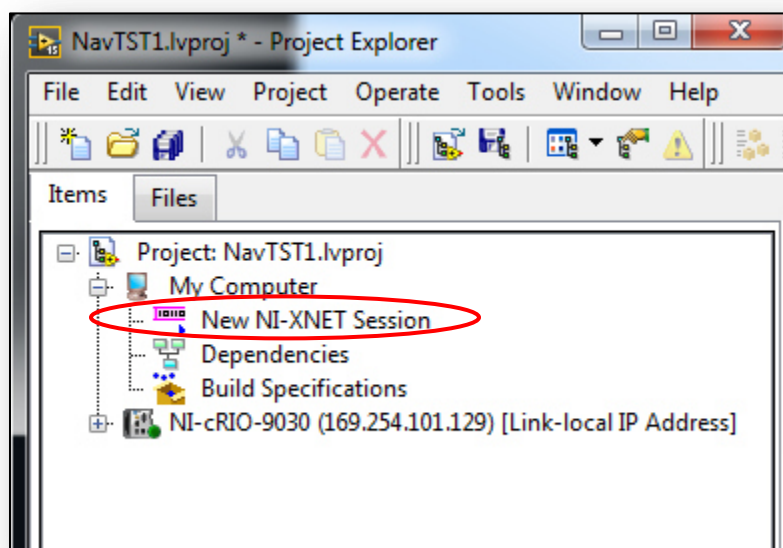


Fig. 29. Sesión creada.

Finalizada la sesión de NI-XNET en LabVIEW ya no es necesario mantener conectado el equipo de adquisición de datos al camión. Esto se hace primeramente para habilitar y sincronizar el módulo con la red CAN pero el resto de la programación que se describirá más adelante se realiza sin conexión.

4.2.4 Desarrollo de código.

Para iniciar con la programación del código, en un nuevo proyecto en LabVIEW se abre un nuevo documento llamado “vi” (virtual instrument), de manera natural, cada “vi” cuenta con dos pantallas en las que se puede programar gráficamente: “Front Panel” (Fig. 31) y “Block Diagram” (Fig. 30), ambas ventanas funcionan en conjunto pero cada una presenta distintas herramientas gráficas.

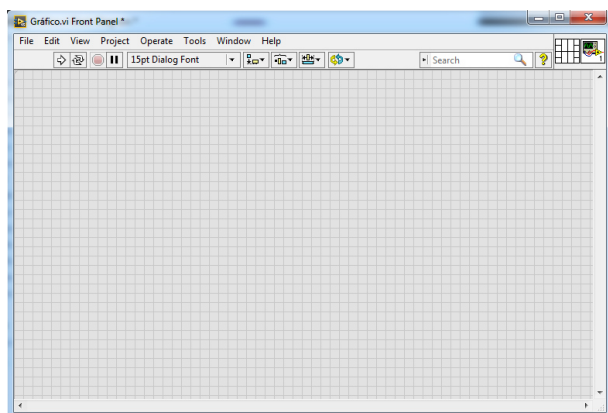


Fig. 31. Front Panel.

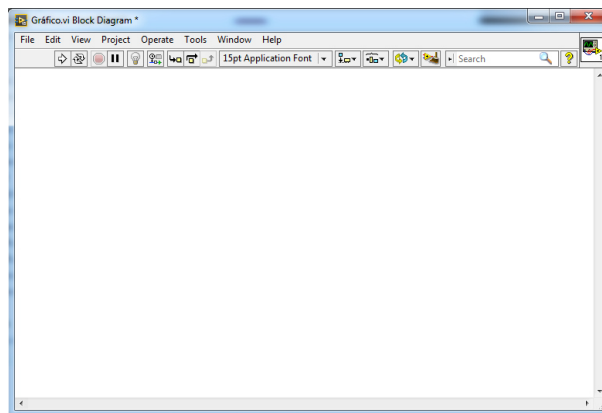


Fig. 30. Block Diagram.

En el “Front Panel” se mostrarán exclusivamente los valores recibidos de la red CAN de manera gráfica. En la ventana “Block Diagram” se realiza todo el proceso correspondiente para cada parámetro, desde cómo es recibido el dato de la sesión NI-XNET hasta que se proyecta en unidades ingenieriles en el panel frontal.

Por cuestiones de Políticas de Privacidad y de propiedad intelectual de la empresa, no se permite mostrar el código de programación tal cual se realizó, pero se mostrará un diagrama general a bloques lo más apegado posible a la realidad.

El desarrollo del código de programación se divide en las siguientes sub-áreas:

- Insertar sesión XNET en virtual instrument.
- Filtrar por nombre de grupo (PGN's).
- Descomponer por byte(s) correspondientes a cada parámetro.
- Realizar ecuaciones de conversión.
- Mostrar en “Front Panel”.
- Almacenar datos en archivo TDMS.

A continuación, apoyado en distintas figuras, se describe cada uno de los pasos que se necesitó para desarrollar el código de programación.

- Insertar Sesión XNET en virtual instrument: Desde el proyecto en la página principal de LabVIEW (Fig. 32) arrastrar la sesión NI-XNET a la ventana de “Block Diagram”, y se deberá mostrar como en la figura 33.

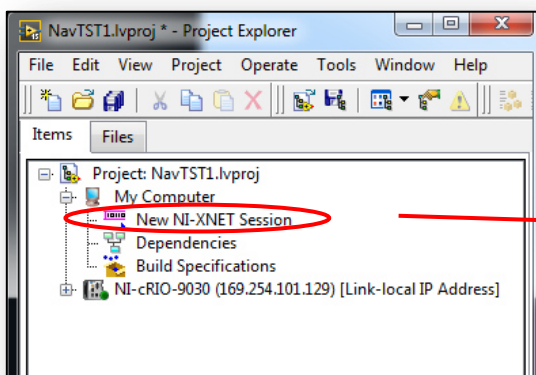


Fig. 32. Sesión creada.

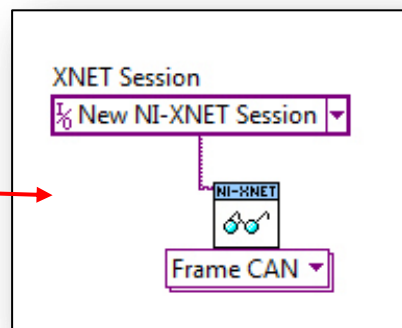


Fig. 33. Sesión NI-XNET en "Block Diagram".

- Filtrar por nombre de grupo (PGN): Una vez que se insertó la sesión, el siguiente paso es filtrar entre cada PGN al que pertenece cada parámetro solicitado, como ya se describió en el apartado **4.2.1. Identificación de parámetros en Norma SAE J1939-71** los PGN's en hexadecimal de cada parámetro es: “F004” para “Engine Speed”, “F003” para “Accelerator Pedal Position”, “FEF1” para “Vehicle Speed” y “FEEEE” para “Engine Coolant Temperature”, en el código de programación se presenta de acuerdo a la figura 34.

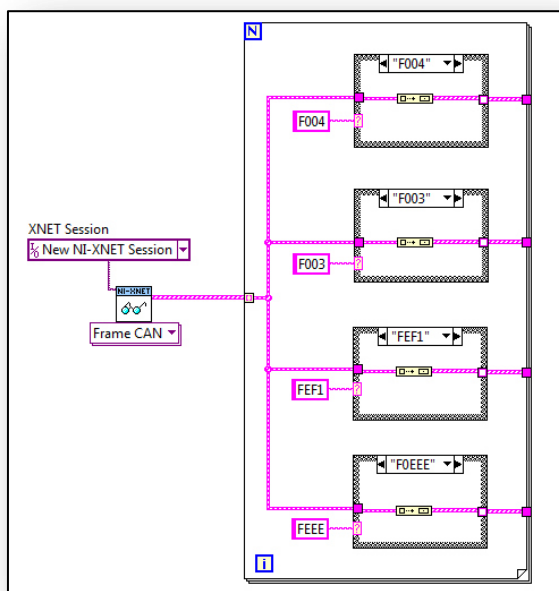


Fig. 34. Filtrado por PGN.

- Descomponer por byte(s) correspondientes a cada parámetro y realizar ecuaciones de conversión: Estas dos sub-áreas, por cuestiones de programación, se hacen en la misma estructura como se muestra en la figura 35. Primero con una herramienta llamada “Index array” se separa de la trama los bytes que se necesitan (de acuerdo al documento de la SAE J1939/71), posteriormente se suman (en caso de que sean más de dos bytes) para finalmente multiplicar el valor tomado por el factor de conversión que también se encuentra en el manual.

El único parámetro que no se multiplica, como se podrá observar en la figura 35, es el “Engine Coolant Temperature” porque a este parámetro corresponde uno a uno el valor que recibimos contra la magnitud, es decir el factor de multiplicación es un grado Celsius por bit.

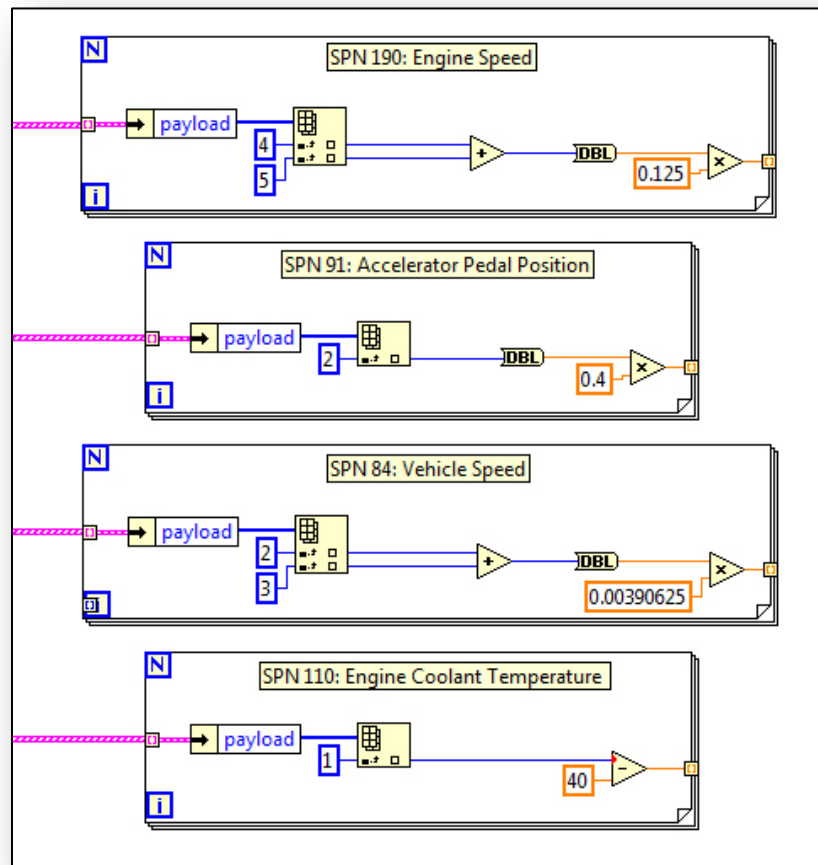


Fig. 35. Conversión de parámetros.

- Mostrar en “Front Panel”: En esta sección se muestran los parámetros ya decodificados, es la única ventana con la que tendrá interacción el usuario. Se compone de dos controles de agujas para la velocidad del vehículo y la

velocidad del motor (RPM), además de una barra de porcentaje para la posición del pedal de aceleración (0-100%) y por último un termómetro para la temperatura del refrigerante en el motor (Fig. 36).

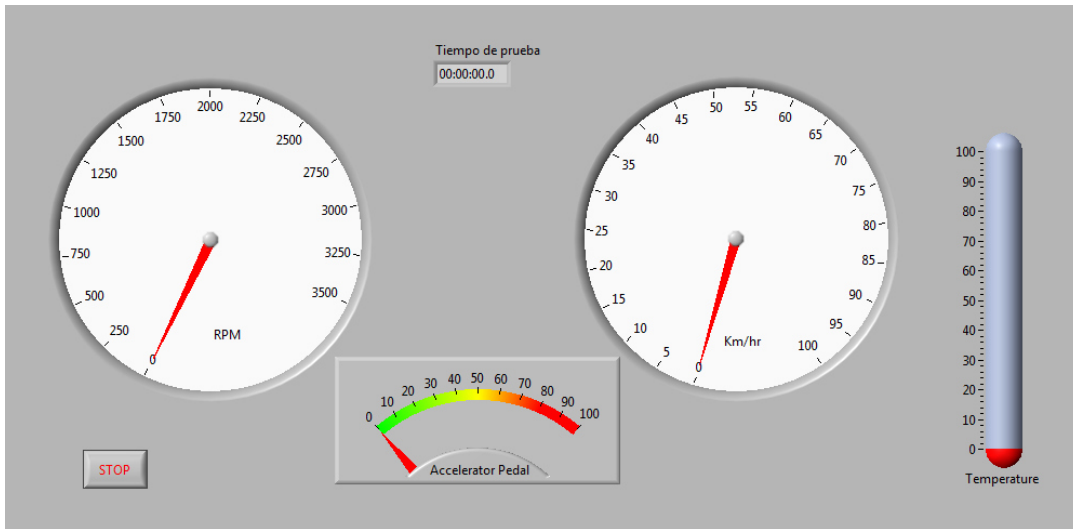


Fig. 36. Panel frontal finalizado.

- Almacenar datos en archivo TDMS: Es la parte final del código (Fig. 37), simultáneamente cada parámetro que se muestra en el “Front Panel” durante una corrida se grabará en la estructura descrita en el Capítulo 3, sección 3.3. La estructura del código establece que una vez que se encuentre conectado el sistema al camión (encendido) y se presione “Run” comenzará a grabar por tiempo determinado y hasta que se presione el botón “Stop” se cerrará el archivo TDMS con sus respectivos datos, la interacción entre el usuario y la aplicación siempre se dará por medio del Host.

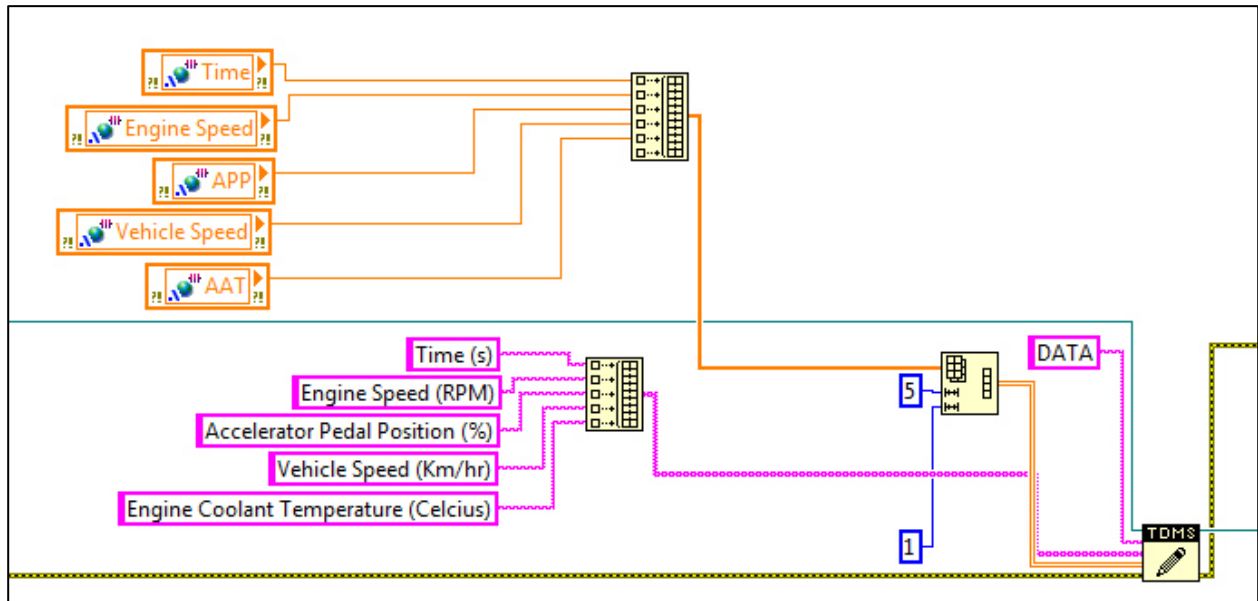


Fig. 37. Código de almacenamiento.

Así se concluye la metodología utilizada para el desarrollo exitoso de la aplicación, en secciones posteriores se describirá la manera adecuada de realizar pruebas en campo con esta aplicación mostrada.

4.3 Discusiones

Con todo lo desarrollado a lo largo de este capítulo se demuestra que la aplicación es buena porque es ligera, amigable y tiene la posibilidad de ampliarse la lectura de los datos recibidos por la ECU del vehículo de una manera muy sencilla gracias a la estructura de programación con que se cuenta. La gran ventaja que presenta sobre equipos como el CANalyzer de Vector radica en que está hecho a medida y se potencializa la función de leer y almacenar ciertos parámetros en magnitudes ingenieriles.

También comparado con el CANalyzer, la aplicación desarrollada, en funciones, es muy pequeña por la diversidad que ofrece el equipo alemán, principalmente porque CANalyzer almacena todos los parámetros existentes en determinado camión a la vez, mientras que la interfaz desarrollada, hasta el momento, solo lee 4. Sin embargo, debido a que almacena todos estos parámetros lo vuelve una herramienta pesada, lenta y compleja al momento de querer visualizar de manera gráfica algún parámetro en específico, esto se demuestra más a detalle en el capítulo 6 de resultados.

Por otra parte, otra función que tiene CANalyzer es la lectura y detección de códigos de diagnóstico de fallas, herramienta con la que no cuenta la interfaz descrita en esta tesis, no obstante, de ser requerido por la empresa, solo con el documento de SAE J1939-73 [14] y el mismo hardware con el que ya se cuenta de National Instruments, en un futuro se tendría la capacidad de ampliar a tal manera de alcanzar esta función. Esta herramienta sirve para detectar fallas en los diversos sub-sistemas con los que cuenta el camión, por ejemplo Power Train, Transmission, Air Bags, ABS, etc.

Capítulo 5: Prueba de concepto

Durante este capítulo quinto se describirá la metodología que se consideró para la validación de la presente aplicación; se describirán las condiciones pertinentes que se tomaron para realizar las pruebas comparativas y evitar factores que alteren la lectura y almacenamiento de los datos CAN. Finalmente se discutirán algunas posibles propuestas de mejora para las posteriores pruebas de campo de acuerdo a las concluidas.

5.1 Introducción

La metodología que se llevó a cabo para validar la interfaz desarrollada es la de comparación con respecto a un equipo validado para su uso en la industria con características similares a las desarrolladas para esta tesis. Se toma como referencia válida los datos adquiridos por CANalyzer y se calcula el porcentaje de error en base a la diferencia que exista en la lectura de CANalyzer y la aplicación desarrollada para este proyecto.

5.2 Metodología

La metodología consistió en tomar un vehículo propiedad del departamento de pruebas de la empresa, conectar los equipos correspondientes a la red CAN y realizar lecturas de los parámetros de la red. Los pasos que se llevaron a cabo para la validación de la aplicación desarrollada se describen a continuación.

5.2.1 Camión de pruebas.

La ficha técnica del vehículo que se utilizó para las pruebas de validación se observa a continuación en la tabla 1.

Marca:	International	Motor:	Diesel 14.9 L cc
Modelo:	LoneStar	Caballos de fuerza:	550
Clase:	Clase 8	Marca de transmisión:	Eaton/Fuller
Marca de motor:	Cummins	Cambios:	18
Frenos ABS:	Si	Cabina:	Convencional
Suspensión:	Aire		

Tabla 1. Ficha técnica

5.2.2 Preparación previa.

Para proceder con la pruebas de validación es importante asegurarse de tener conexiones correctas desde alimentación de los equipos hasta con la interfaz del conector del vehículo. A continuación y apoyado con imágenes se muestra detalladamente cada paso necesario para lograr una exitosa preparación para finalmente realizar las pruebas en pista.

- *Alimentación.*

Primeramente se debe alimentar el procesador de datos CRIO-9030 de National Instruments a las baterías del camión 12 VDC.

Cabe recalcar que solo el funcionamiento de nuestra aplicación requiere alimentación directa de las baterías, por su cuenta, el equipo de VECTOR se alimenta automáticamente también con 12 VDC al enlazarse al conector J1939 del camión.

- *Unidad de almacenamiento.*

El procesador NI CRIO-9030 cuenta con una memoria interna de un GB en donde se podría almacenar sin problema el archivo de lectura de parámetros, sin embargo para esta aplicación del sistema nos es de mayor utilidad insertar una memoria USB de almacenamiento en uno de sus puertos y dirigir el archivo a esta dirección para mayor facilidad a su acceso, solo se retira la unidad y se inserta en un puerto USB de una computadora para análisis [6].

De acuerdo a la estructura de programación y para evitar no almacenar una prueba en campo, no se permite iniciar una lectura de parámetros sin una unidad de almacenamiento, si esto se intentara surge una ventana en donde te indica la falta de memoria en el puerto y no te permite continuar hasta que sea insertada. A su vez, ya iniciada la aplicación se abre un archivo TDMS y lo cierra una vez que se detuvo el funcionamiento, la estructura y contenido de este presente archivo se observará y discutirá en el capítulo 6.

- *Conexión: Host - Procesador.*

Ya alimentado el procesador, se procede a conectar dicho procesador al host por medio de un cable Ethernet para establecer comunicación entre la aplicación desarrollada en LabVIEW y el CRIO-9030. Como se ha explicado en capítulos anteriores, la conexión se establece a través del software NI MAX.

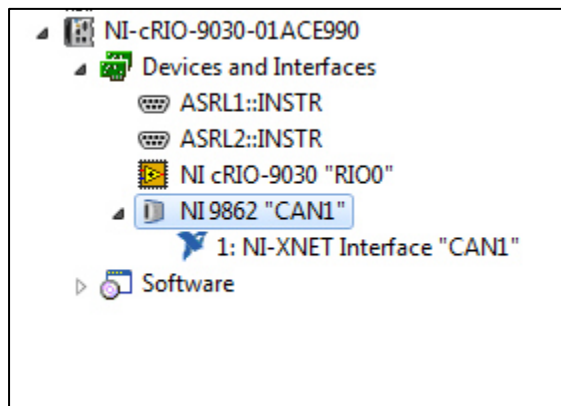


Fig. 38. Conexión NI MAX.

- *Conexión: vehículo - equipos DAQ.*

La última sección de este apartado consiste en establecer comunicación entre ambos sistemas DAQ (NI y CANalyzer) en paralelo a la red CAN del vehículo, como se puede observar en la figura 41, apoyado en el cable de dos vías del conector J1939; se conecta un extremo del cable a la interfaz de vehículo, localizada en la parte inferior izquierda del tablero, y en el otro extremo permite tener dos conexiones (fig. 41) disponibles para cada sistema que se enlazará al vehículo: por medio de un cable J1939-USB para CANalyzer y J1939-Serial para National Instruments.



Fig. 39. Conexión a la red CAN del vehículo.

5.2.3 Toma de lecturas.

Para realizar la validación de esta aplicación se tomaron diversas lecturas, con un promedio de 5 minutos cada una, con ambos sistemas, esto permitirá recolectar una cantidad considerable de datos para posteriormente comparar la información de los parámetros por medio de los dos equipos. En la figura 42 se presenta de manera gráfica la metodología aplicada en la toma de lecturas.

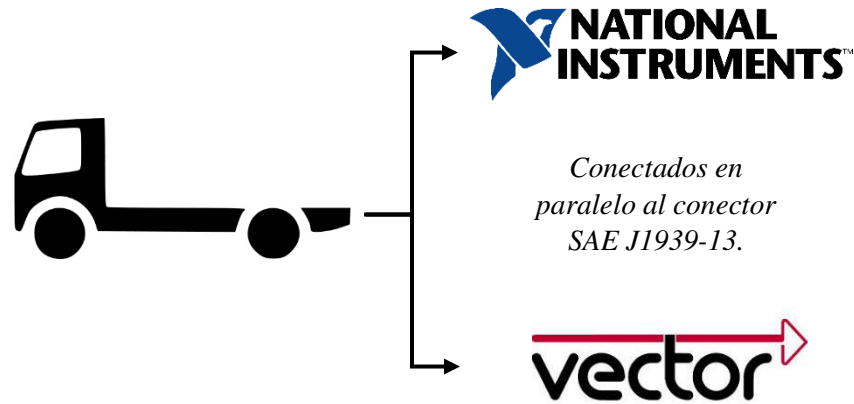


Fig. 40. Estructura de validación.

De acuerdo a la estructura de programación que se desarrolló en esta aplicación, al ejecutarse, automáticamente iniciará a almacenar los 4 parámetros de los que se programó para que filtrara (Engine Speed, Accelerator Pedal Position, Vehicle Speed y Engine Coolant Temperature). Con apoyo en una flecha color rojo de la figura 43 se puede observar el botón de “Run” en LabVIEW para iniciar con la lectura.



Fig. 41. Interfaz de usuario.

A diferencia de la interfaz desarrollada en National Instruments, CANalyzer muestra al usuario todo el conjunto de parámetros ordenados por PGN, como se puede observar en la figura 44. Una vez que se inicia el almacenamiento con este software, graba todos los parámetros recibidos a la vez; el filtrado de los 4 parámetros interesados se da en el procesamiento posterior de manera manual. Cómo se obtuvieron estas magnitudes será descrito detalladamente en la sección 5.3.2.

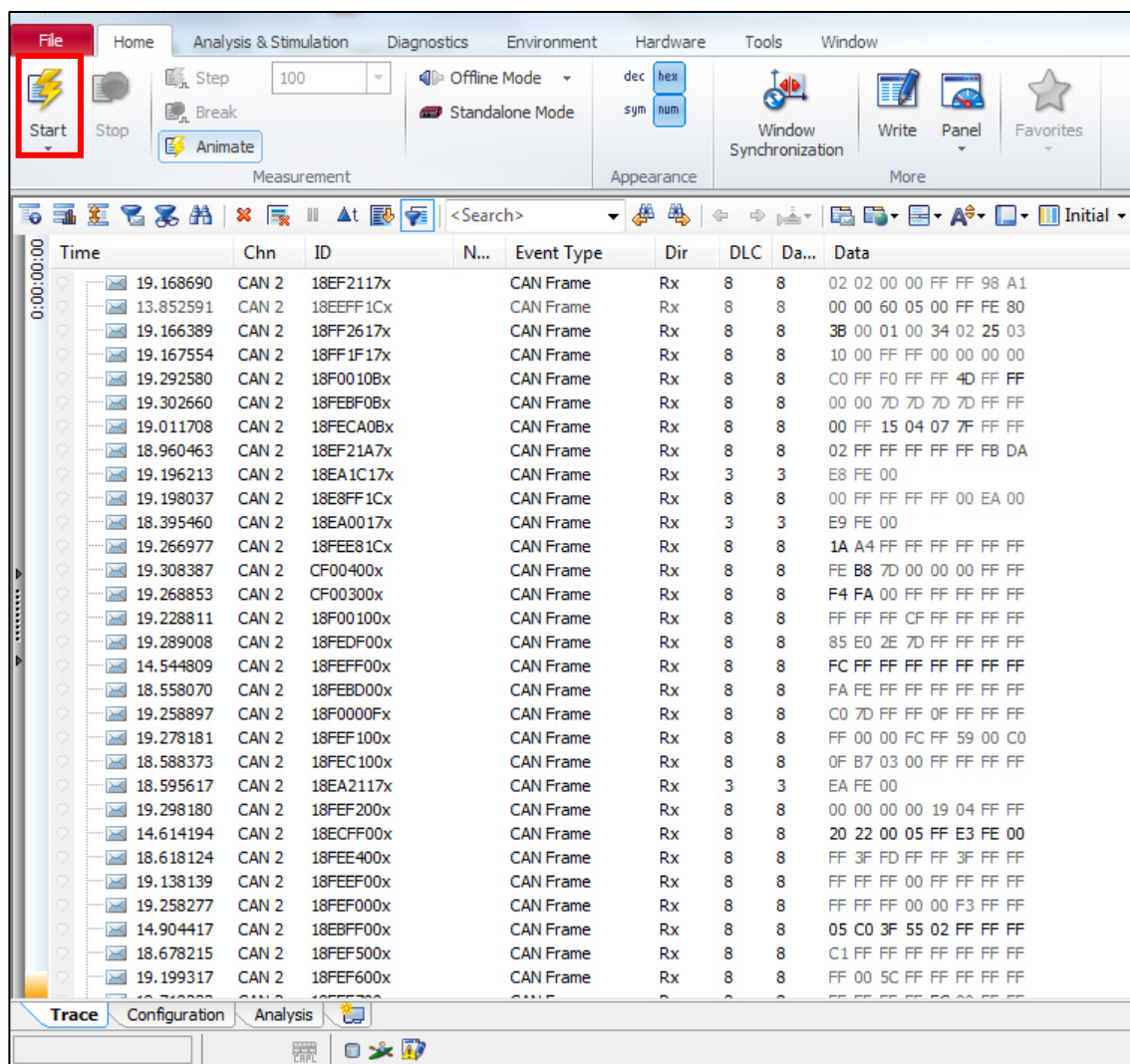


Fig. 42. Interfaz CANalyzer.

Para la toma de lecturas con CANalyzer se recibió amable apoyo del personal capacitado por la empresa para el uso del software y adquirir correctamente los datos de almacenamiento. Estas pruebas se llevaron a cabo en instalaciones internas de la empresa, siguiendo rutas sin un patrón específico, debido a que lo que se busca comparar es indiferente al trayecto recorrido, pero sí al comportamiento interno del vehículo y cada sub sistema.

5.3 Resultados

En los resultados recolectados en las diversas lecturas se muestran 2 distintas vertientes, por una lado se tiene la aplicación desarrollada en National Instruments en donde los archivos se guardan en una unidad de almacenamiento USB y basta con insertarla en una computadora con Excel para poder observar sus magnitudes, obtener gráficas y tener la libertad de analizarlas. Por otra parte con el software CANalyzer se

vuelve considerablemente más complejo acceder a las magnitudes recolectadas, en los siguientes apartados se mostrará detalladamente cómo se obtuvieron los resultados.

5.3.1 Interfaz desarrollada bajo hardware de National Instruments.

La interfaz que se desarrolló, fue programada para que por cada lectura genere un archivo TDMS (editable en Excel) con el nombre de TotalDATA.x, en donde “x” representa el número de lectura en curso para todas las pruebas necesarias

De acuerdo a la programación, todos los archivos TDSM que se generan presentan una misma estructura en el contenido, por ejemplo: dos pestañas de Excel en donde en la primera se puede observar el nombre del archivo, la cantidad de canales, el nombre de cada columna (parámetro CAN) y la cantidad de datos almacenados de cada uno; como se muestra en la siguiente figura 45.

Root Name	Title	Author	Date/Time	Groups	Description			
TotalDATA1.1				1				
Group	Channels	Description						
DATA	5							
DATA								
Channel	Datatype	Unit	Length	Minimum	Maximum	Description	NI_ArrayColumn	Start Index
Time (s)	DT_DOUBLE		200389					0
Engine Speed (RPM)	DT_DOUBLE		200389					1
Accelerator Pedal Position (%)	DT_DOUBLE		200389					2
Vehicle Speed (Km/hr)	DT_DOUBLE		200389					3
Engine Coolant Temperature (Celcius)	DT_DOUBLE		200389					4

Fig. 43. Primer pestaña.

En la pestaña número 2 se muestran los datos recolectados de cada parámetro en sus respectivos tiempos divididos por columnas, como se observa en la figura 45, de manera muy sencilla, ordenada y en unidades ingenieriles. Así, se facilita analizar por medio de gráficas los resultados de cada parámetro.

	A	B	C	D	E	F
1	Time (s)	Engine Speed (RPM)	Accelerator Pedal Position (%)	Vehicle Speed (Km/hr)	Engine Coolant Temperature (Celcius)	
2	0	770.75	0	0	68	
3	0	770.75	0	0	68	
4	0.001949787	770.75	0	0	68	
5	0.003915787	770.75	0	0	68	
6	0.005906582	770.75	0	0	68	
7	0.007905006	770.75	0	0	68	
8	0.009905815	770.75	0	0	68	
9	0.011952877	770.75	0	0	68	
10	0.013904572	770.75	0	0	68	
11	0.015902996	770.75	0	0	68	
12	0.017903805	770.75	0	0	68	
13	0.019906044	770.75	0	0	68	
14	0.021901608	770.75	0	0	68	
15	0.023950577	770.75	0	0	68	
16	0.025905609	770.75	0	0	68	
17	0.027904034	770.75	0	0	68	
18	0.029903889	770.75	0	0	68	
19	0.031900883	770.75	0	0	68	
20	0.033903599	770.75	0	0	68	
21	0.035953045	770.75	0	0	68	
22	0.0379076	770.75	0	0	68	
23	0.039903641	770.75	0	0	68	
24	0.041903610	770.75	0	0	68	

Fig. 44. Segunda Pestaña.

Para graficar un parámetro se selecciona la columna completa de interés, se presiona la primera opción de la sección “Insert Line Chart” como se muestra en la figura 47.

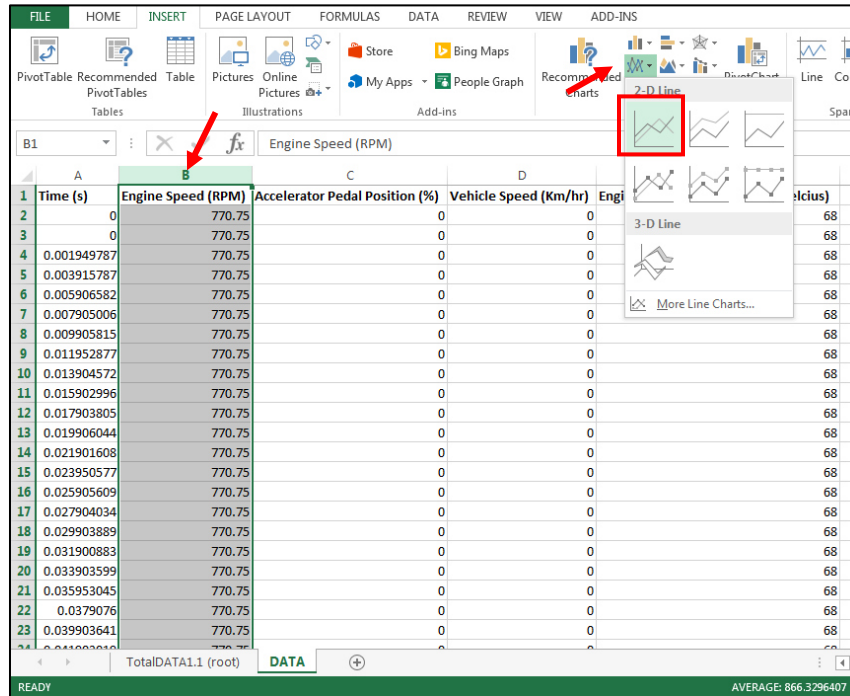


Fig. 45. Selección "Engine Speed".

Al realizar lo anteriormente descrito se obtiene la gráfica que se muestra en la figura 48.

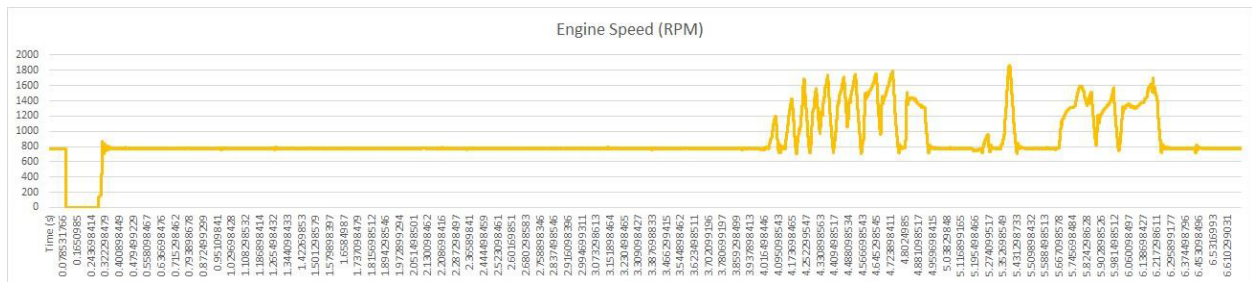


Fig. 46. Engine Speed.

De esta misma manera se seleccionan y grafican los 3 parámetros restantes de esta lectura, con lo que se obtiene lo siguiente

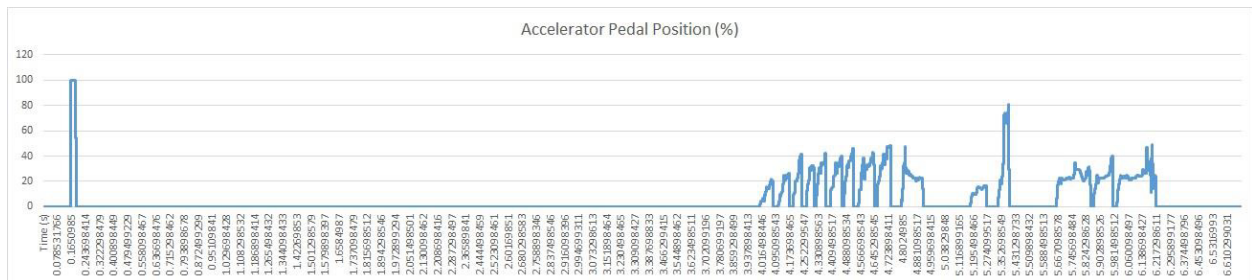


Fig. 47. Accelerator Pedal Position.

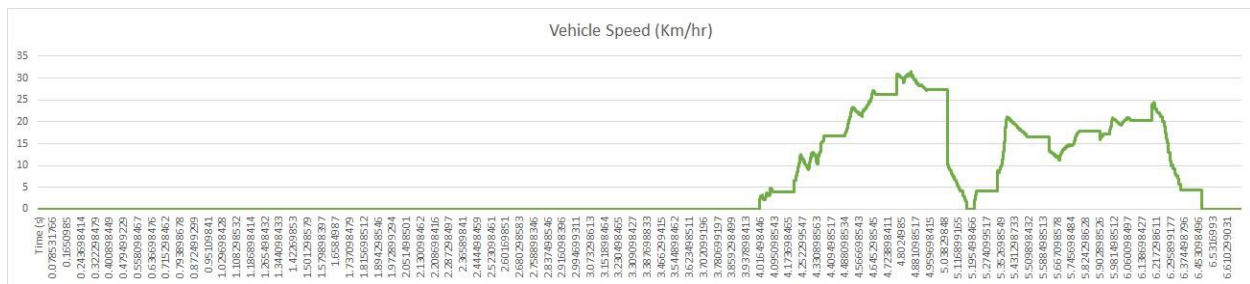


Fig. 48. Vehicle Speed.

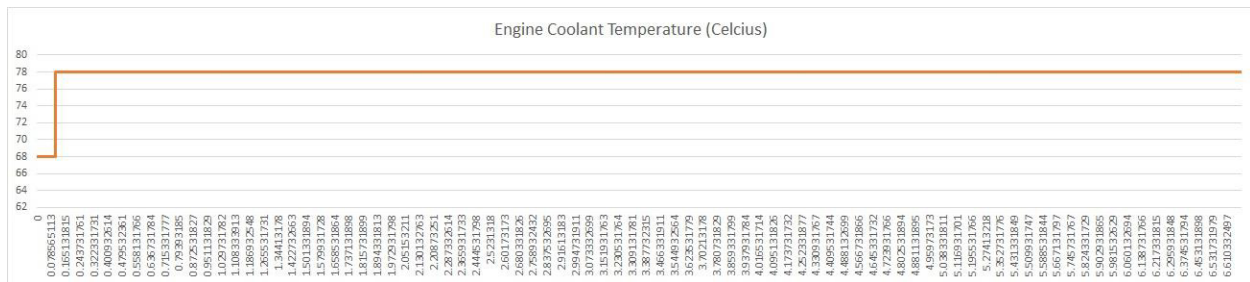


Fig. 49. Engine Coolant Temperature.

5.3.2 Software CANalyzer.

Cuando en Vector se inicia una lectura de parámetros CAN, SAE J1939 en este caso, de manera natural se graba en un archivo extensión “.blf”. Este archivo nos permite exportarlo a una extensión de archivo más amigable para poder analizarlo debidamente y compararlo con los datos obtenidos con National Instruments. Para efectos de esta comparación, se exporta a un archivo “.asc” y así tener la libertad de abrirlo y editarlo en Excel.

Como se muestra en la figura 51, al dar clic derecho en cualquier parámetro de la lectura se abre un menú de opciones de donde se elige “Import/Export” y a su vez en la opción “Export”.

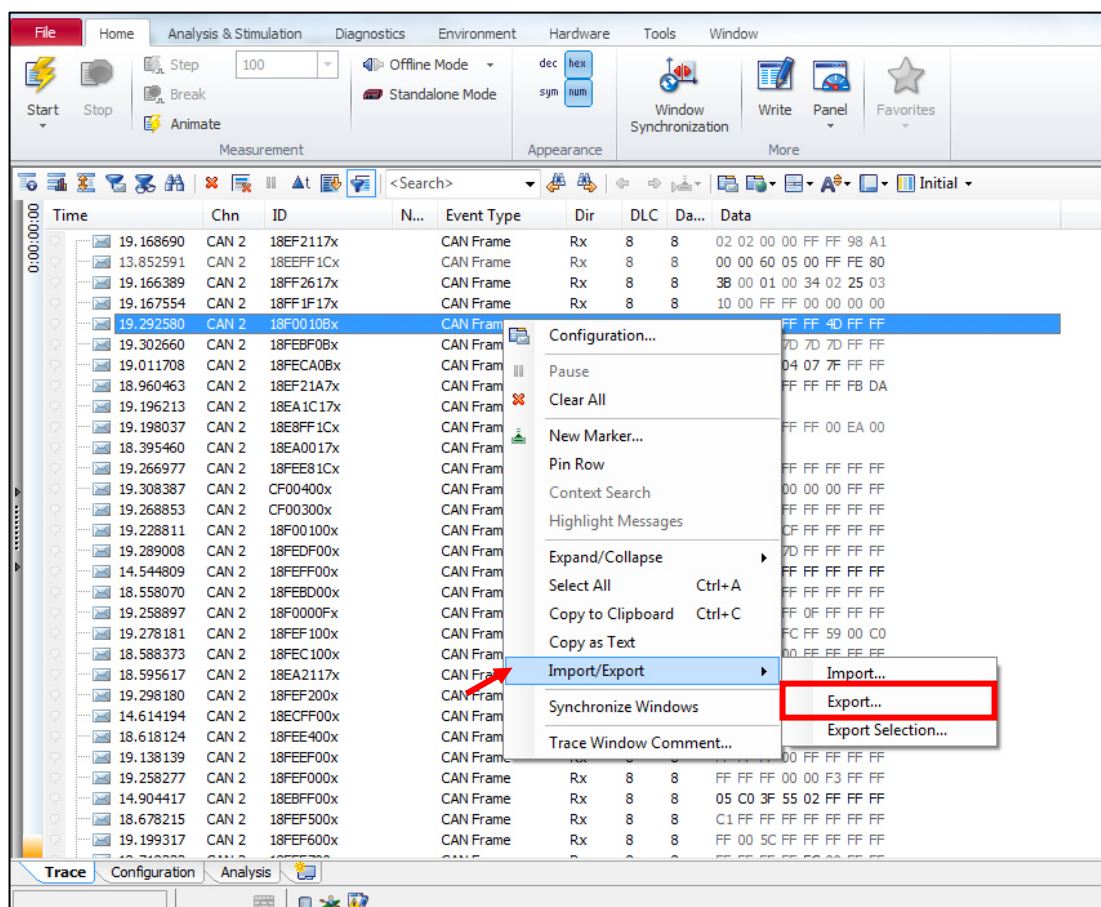


Fig. 50. CANalyzer exportación de archivo.

En la ventana emergente que se abrirá en seguida aparecerá el destino final del archivo y su extensión en “.blf” (Fig. 52), sin embargo en esta sección se puede cambiar a la extensión de archivo deseada, si se presiona el recuadro gris que aparece a la derecha (marcado con una flecha) se tendrá la oportunidad de elegir otra extensión de archivo.

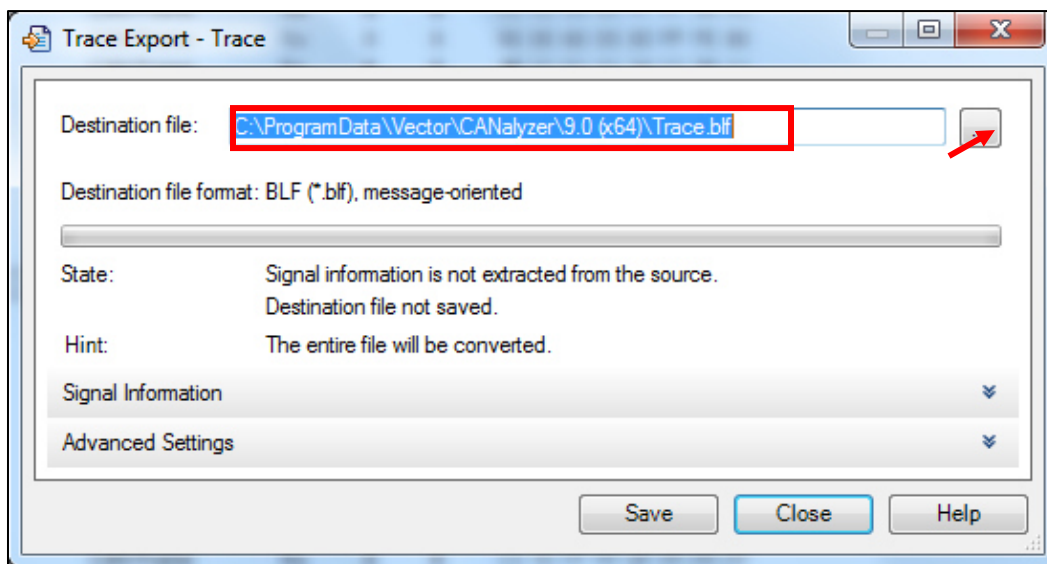


Fig. 51. Trace Export.

Al presionar el recuadro gris aparecerá una ventana como la figura 53, en la que se elige el folder en donde se desea guardar el archivo y la extensión, en este caso la extensión “.asc”.

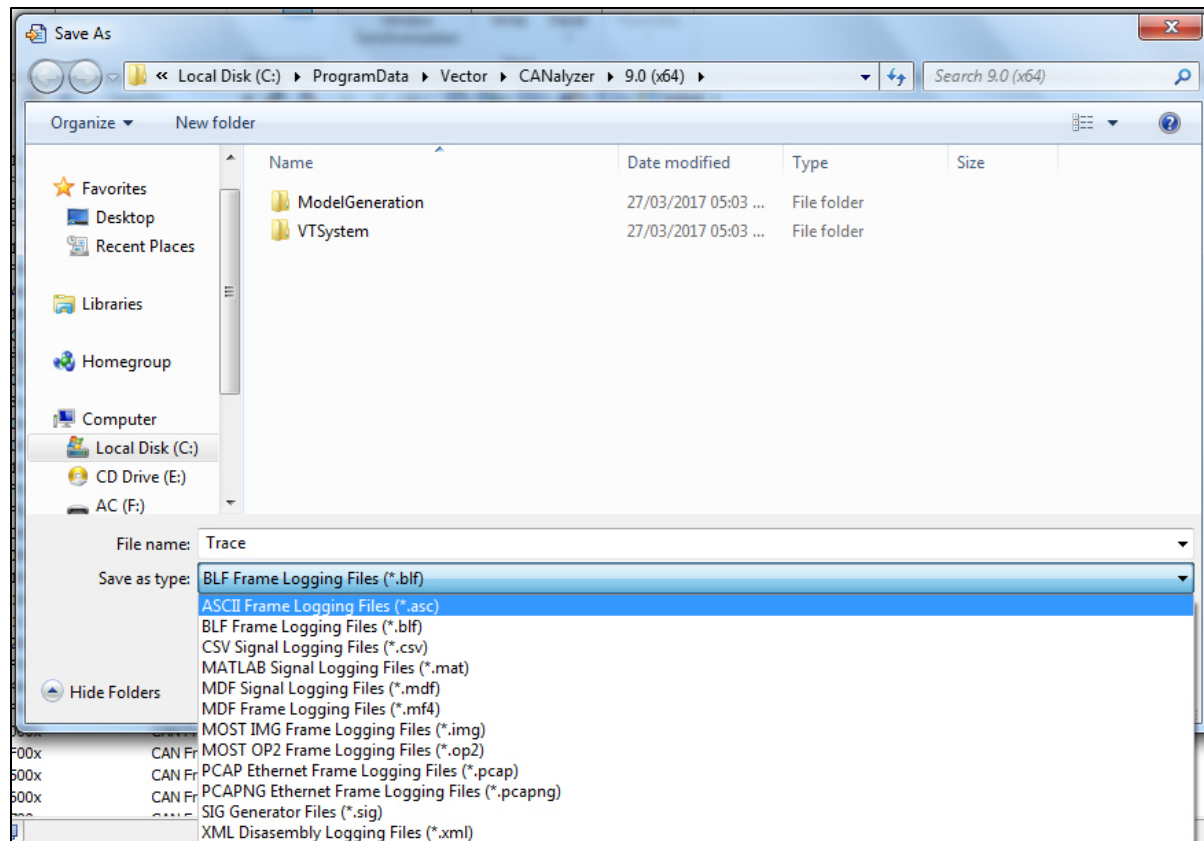


Fig. 52. Guardar archivo.

Finalmente se podrá abrir el archivo en una hoja de Excel, la estructura de este archivo se muestra en la figura 54, en donde se puede observar que todos los parámetros están representados en sistema hexadecimal. Este archivo contiene, entre otras cosas, el tiempo específico de lectura (columna A), el PGN del parámetro (columna C) y los 8 bits que contiene ese parámetro (columnas F a M).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	date Thu Feb 9		14:16.0	m 2017																				
2	base hex	imes	tamps absolute																					
3	internal e	vents	logged																					
4	// version 8.5.		0																					
5	0.034546	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
6	1.034546	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
7	2.034546	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
8	3.034546	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
9	4.034547	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
10	5.034547	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
11	6.034547	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
12	7.034548	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
13	8.034548	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
14	9.034548	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
15	10.03455	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
16	11.03455	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
17	12.03455	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
18	13.03455	CAN	2 Status:chip	status		er	ror	ac	tiv	e														
19	13.82064		2 18EF2117x	Rx	d	8	2	2	0	0	FF	FF	98	A1 Length =	548000	BitCount =	141	ID =	418324759x					
20	13.83824		2 18EF2117x	Rx	d	8	2	2	0	0	FF	FF	98	A1 Length =	548000	BitCount =	141	ID =	418324759x					
21	13.85259		2 18EEFF1Cx	Rx	d	8	0	0	60	5	0	FF	FE	80 Length =	568000	BitCount =	146	ID =	418316060x					
22	13.95829		2 18FF2617x	Rx	d	8	22	0	0	0	5C	1	23	3 Length =	548000	BitCount =	141	ID =	419374615x					
23	13.95887		2 18FF1F17x	Rx	d	8	2	0	FF	FF	0	0	0	0 Length =	568000	BitCount =	146	ID =	419372823x					
24	13.96373		2 18FF1F17x	Rx	d	8	2	0	FF	FF	0	0	0	0 Length =	568000	BitCount =	146	ID =	419372823x					

Fig. 53. Archivo .asc de CANalyzer.

Realizando el filtrado correspondiente y recordando la información de cada parámetro que nos proporciona el documento SAE-J1939-71, misma que se puntualizó en la sección 4.2.1 Identificación de parámetros en Norma SAE J1939-71, obtenemos lo que se muestra a continuación en el siguiente listado (Tabla 2) de PGN's para proceder al filtrado manual en el documento obtenido.

Parámetro	PGN	Bytes	Resolución por bit
Engine Speed	F004	4-5	0.125 RPM
Accelerator Pedal Position	F003	2	0.4 %
Vehicle Speed	FEF1	2-3	1/256 Km/hr
Engine Coolant Temperature	FEFE	1	1 centígrado

Tabla 2. Parámetros SAE J1939-71

Posteriormente y ya separado los PGS's de los 4 parámetros a analizar se obtiene un nuevo documento (Figura 55) en donde lo correspondiente es, de acuerdo a los bits que especifica la tabla 2 de cada parámetro, multiplicar por su respectiva resolución para obtener magnitudes ingenieriles.

FILE

HOME

INSERT

PAGE LAYOUT

FORMULAS

DATA

REVIEW

VIEW

ADD-INS

Cut

Copy

Paste

Format Painter

Clipboard

Calibri

11

A

A

B

I

U

Font

Alignment

Wrap Text

Merge & Center

Number

General

\$

%

,

←0.00

→0.00

Conditional Formatting

X49

✕

✓

fx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
1		CAN	2 Status	status	er	ror	ac	tiv	e																
2	14.50466	2	CF00400x	Rx	d	8	FF	7D	7D	0	0	0	FF	FF	Length	=	560000	BitCount	=	144	ID	=	217056256x		
3	14.52342	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
4	14.54417	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
5	14.56358	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
6	14.58348	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
7	14.6035	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
8	14.62352	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
9	14.64428	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
10	14.6636	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
11	14.6837	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
12	14.70372	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
13	14.72362	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
14	14.7444	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
15	14.76366	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
16	14.78376	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
17	14.80381	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
18	14.82384	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
19	14.84453	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
20	14.86376	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
21	14.88379	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
22	14.9038	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
23	14.92394	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
24	14.94418	2	CF00400x	Rx	d	8	FE	7D	7D	0	0	0	FF	FF	Length	=	564000	BitCount	=	145	ID	=	217056256x		
		F004	F003	FEBF	FEEE																				

READY

Fig. 54. Filtrado por PGN.

De esta manera se calcula el valor de cada parámetro almacenado, tal y como se puede observar en la figura 56 para finalmente graficar el conjunto de datos.

FILE		HOME		INSERT		PAGE LAYOUT		FORMULAS		DATA		REVIEW		VIEW		ADD-INS								
<div><div><div>Cut</div><div>Copy</div><div>Paste</div><div>Format Painter</div></div></div>		<div><div>Calibri</div><div>11</div><div>A⁺</div><div>A⁻</div></div>		<div><div>≡</div><div>≡</div><div>≡</div></div>		<div><div>Wrap Text</div><div>Merge & Center</div></div>		<div><div>General</div><div>\$</div><div>%</div><div>,</div><div>←.00</div><div>→.00</div></div>		<div><div>Conditional Formatting</div><div>Table</div></div>														
Clipboard		Font		Alignment		Number		Styles																
AH3076																								
	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	Z	AA
1	CAN	2 Status:cl	status	er	ror	ac	tiv	e															MAGNITUD	
3068	2	CF00400x	Rx	d	8	F0	7D	88	29	18	0	FF	FF	Length	=	560000	BitCount	=	144	ID	=	217056256x	773.125	RPM
3069	2	CF00400x	Rx	d	8	F0	7D	88	35	18	0	FF	FF	Length	=	556000	BitCount	=	143	ID	=	217056256x	774.625	RPM
3070	2	CF00400x	Rx	d	8	F0	7D	88	3C	18	0	FF	FF	Length	=	560000	BitCount	=	144	ID	=	217056256x	775.5	RPM
3071	2	CF00400x	Rx	d	8	F0	7D	89	29	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	773.125	RPM
3072	2	CF00400x	Rx	d	8	F0	7D	88	24	18	0	FF	FF	Length	=	556000	BitCount	=	143	ID	=	217056256x	772.5	RPM
3073	2	CF00400x	Rx	d	8	F0	7D	88	33	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	774.375	RPM
3074	2	CF00400x	Rx	d	8	F0	7D	88	31	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	774.125	RPM
3075	2	CF00400x	Rx	d	8	F0	7D	89	18	18	0	FF	FF	Length	=	556000	BitCount	=	143	ID	=	217056256x	771	RPM
3076	2	CF00400x	Rx	d	8	F0	7D	88	0F	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	769.875	RPM
3077	2	CF00400x	Rx	d	8	F0	7D	88	1B	18	0	FF	FF	Length	=	556000	BitCount	=	143	ID	=	217056256x	771.375	RPM
3078	2	CF00400x	Rx	d	8	F0	7D	88	2B	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	773.375	RPM
3079	2	CF00400x	Rx	d	8	F0	7D	89	1B	18	0	FF	FF	Length	=	548000	BitCount	=	141	ID	=	217056256x	771.375	RPM
3080	2	CF00400x	Rx	d	8	F0	7D	88	2B	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	773.375	RPM
3081	2	CF00400x	Rx	d	8	F0	7D	88	3C	18	0	FF	FF	Length	=	560000	BitCount	=	144	ID	=	217056256x	775.5	RPM
3082	2	CF00400x	Rx	d	8	F0	7D	88	4B	18	0	FF	FF	Length	=	548000	BitCount	=	141	ID	=	217056256x	777.375	RPM
3083	2	CF00400x	Rx	d	8	F0	7D	89	2C	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	773.5	RPM
3084	2	CF00400x	Rx	d	8	F0	7D	88	29	18	0	FF	FF	Length	=	560000	BitCount	=	144	ID	=	217056256x	773.125	RPM
3085	2	CF00400x	Rx	d	8	F0	7D	87	4E	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	777.75	RPM
3086	2	CF00400x	Rx	d	8	F0	7D	88	42	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	776.25	RPM
3087	2	CF00400x	Rx	d	8	F0	7D	88	4A	18	0	FF	FF	Length	=	548000	BitCount	=	141	ID	=	217056256x	777.25	RPM
3088	2	CF00400x	Rx	d	8	F0	7D	87	75	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	782.625	RPM
3089	2	CF00400x	Rx	d	8	F0	7D	87	82	18	0	FF	FF	Length	=	552000	BitCount	=	142	ID	=	217056256x	784.25	RPM
3090	2	CF00400x	Rx	d	8	F0	7D	88	7A	18	0	FF	FF	Length	=	548000	BitCount	=	141	ID	=	217056256x	783.125	RPM
F004 F003 FEBF FEEF +																								
READY																								

Fig. 55. Magnitud de Engine Speed.

Realizando esta metodología se obtienen las gráficas de las magnitudes de los 3 parámetros restantes de esta lectura. Estas gráficas se muestran a continuación en la figuras 57 a 60.

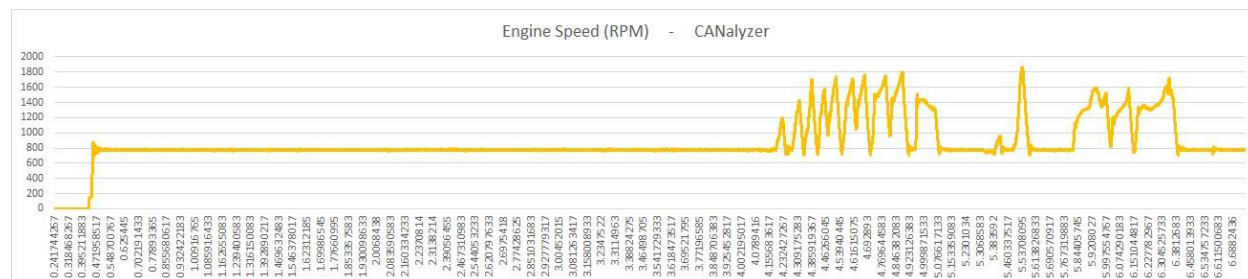


Fig. 56. Engine Speed – CANalyzer.

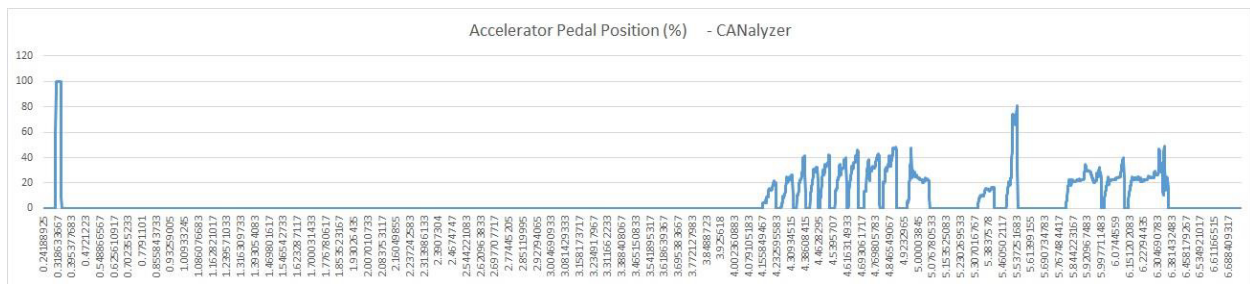


Fig. 57. Accelerator Pedal Position – CANalyzer.

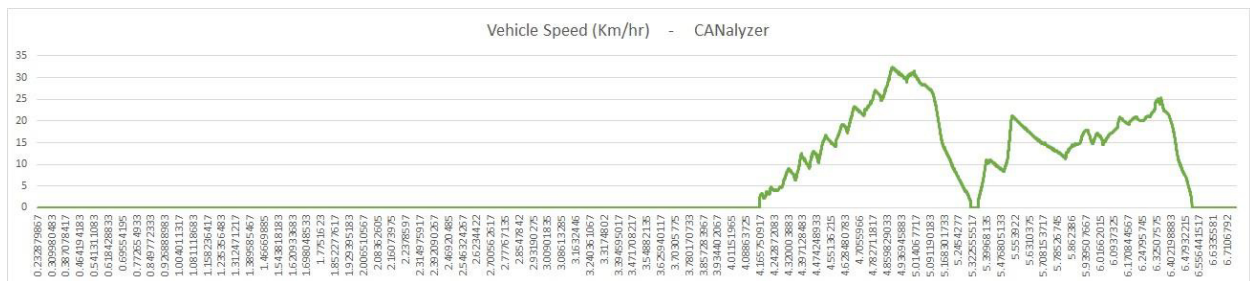


Fig. 58. Vehicle Speed – CANalyzer.

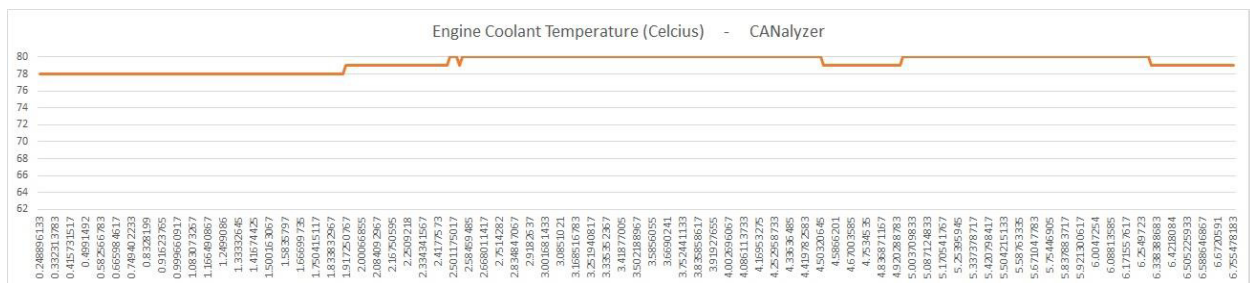


Fig. 59. Engine Coolant Temperature – CANalyzer.

5.3.3. Comparación de resultados.

Para facilitar la comparación entre los datos obtenidos se muestra las gráficas correspondientes a cada parámetro de la lectura antes descrita.

Primeramente se encuentra el resultado de las revoluciones del motor (Engine Speed), en la parte superior a esta imagen se muestran los datos obtenidos por la interfaz desarrollada en este proyecto de tesis y en la parte inferior los datos obtenidos por el software de CANalyzer.



Fig. 60. Resultados comparativos "Engine Speed".

De la misma manera se muestra en la siguiente figura los resultados obtenidos por ambos sistemas de adquisición de los datos del parámetro del porcentaje del pedal de aceleración “Accelerator Pedal Position”.

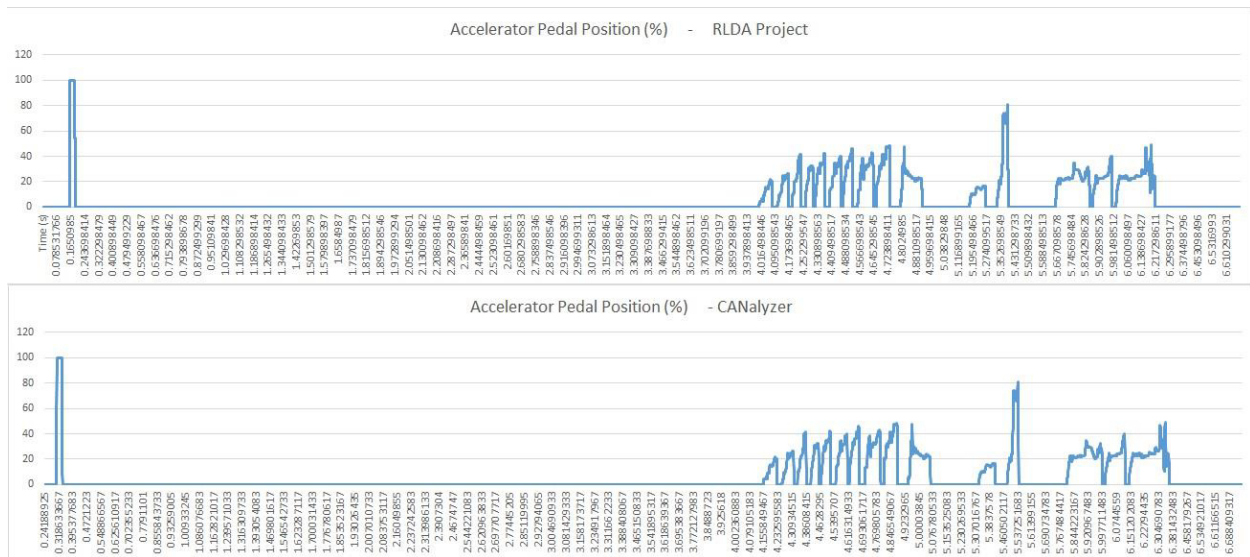


Fig. 61. Resultados comparativos "Accelerator Pedal Position".

Resultados obtenidos de la velocidad del vehículo “Vehicle Speed” (figura 63).

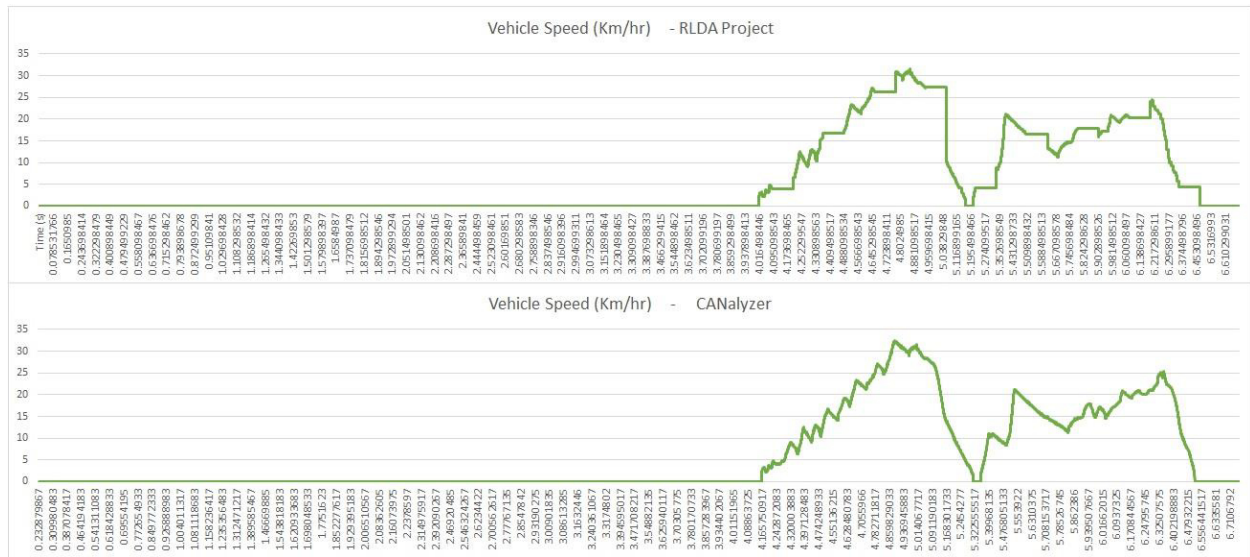


Fig. 62. Resultados comparativos "Vehicle Speed".

Finalmente los resultados del parámetro de temperatura del refrigerante del motor “Engine Coolant Temperature” (Fig. 64).

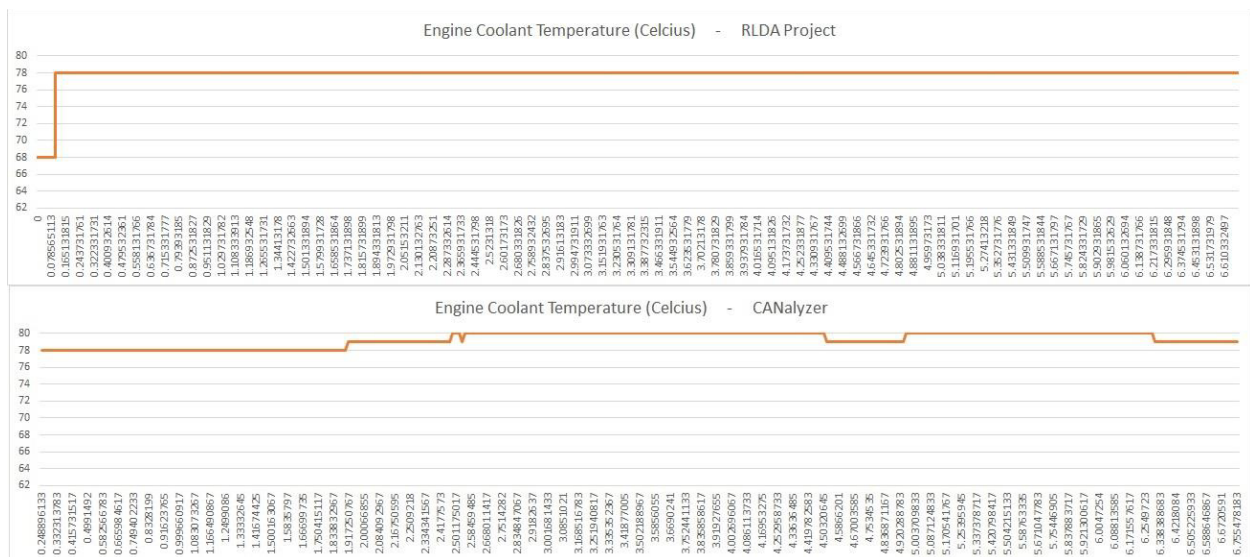


Fig. 63. Resultados comparativos Engine Coolant Temperature.

5.3.4. Porcentaje de error.

Considerando que los datos adquiridos por CANalyzer son los correctos, se realizó una comparación y calculó el porcentaje de error con la siguiente fórmula (1).

$$PE = \left(\frac{|DC - DNI|}{DC} \right) * 100$$

En donde:

PE = Porcentaje de error

DC = DatoCANalyzer

DNI = Dato National Instruments

De manera manual, se calcula en Excel el porcentaje de error de cada punto adquirido por ambos sistemas, después se calcula un promedio de error de todos los datos adquiridos durante los 5 minutos de lectura, como lo muestra la figura 65.

X	Y	Z	AA	AB	AC	AD	AE	AF	AG
	BIN	DEC	MAG CANalyzer	Time	MAG NI	Difference	%Error	Average	
	111100	28	11,2	4,41529	12,8	1,6	14%	8%	
	100000	32	12,8	4,416115	12,8	0	0%		
	100000	32	12,8	4,416964	13,2	0,4	3%		
	100001	33	13,2	4,417782	13,2	0	0%		
	100001	33	13,2	4,418629	13,2	0	0%		
	100110	38	15,2	4,419449	16,8	1,6	11%		
	101010	42	16,8	4,420297	18,4	1,6	10%		
	101110	46	18,4	4,421118	18,4	0	0%		
	101110	46	18,4	4,421963	18,4	0	0%		
	101111	47	18,8	4,422789	20	1,2	6%		
	110010	50	20	4,423632	21,2	1,2	6%		
	110101	53	21,2	4,424455	21,2	0	0%		
	110101	53	21,2	4,4253	21,2	0	0%		
	110101	53	21,2	4,426123	21,2	0	0%		
	110101	53	21,2	4,42697	25,2	4	19%		
	111001	57	22,8	4,427791	25,2	2,4	11%		
	111111	63	25,2	4,428637	25,2	0	0%		
	1000110	70	28	4,429462	29,2	1,2	4%		
	1001001	73	29,2	4,430307	30,8	1,6	5%		
	1001100	76	30,4	4,431128	30,8	0,4	1%		
	1001101	77	30,8	4,431973	30,8	0	0%		
	1001101	77	30,8	4,432797	30,8	0	0%		
	1001101	77	30,8	4,433644	30,8	0	0%		
	1001101	77	30,8	4,434465	30,8	0	0%		
	1001101	77	30,8	4,435317	30,8	0	0%		
	1001101	77	30,8	4,436135	30,4	0,4	1%		
	1001100	76	30,4	4,436979	28,8	1,6	5%		
	1001001	73	29,2	4,437801	28,8	0,4	1%		
	1001000	72	28,8	4,438647	28,8	0	0%		
	1001000	72	28,8	4,439473	28,4	0,4	1%		
	1000111	71	28,4	4,440321	29,2	0,8	3%		
	1001000	72	28,8	4,441138	29,2	0,4	1%		
	1001001	73	29,2	4,441988	32,8	3,6	12%		

Fig. 64. Calculo de porcentaje de error.

De la misma manera se calculan los porcentajes de error de los parámetros y nos arroja los resultados presentados en la tabla 3.

Tabla 3. Porcentajes de error.

Parámetro	Porcentaje de error (%)
Engine Speed	3
Accelerator Pedal Position	8
Vehicle Speed	15
Engine Coolant Temperature	3

5.5 Discusiones

De acuerdo a las pruebas de validación desarrolladas, se llega a la conclusión que es sumamente crítico una correcta conexión entre la interfaz J1939 del camión y el sistema de adquisición de datos para el almacenamiento de datos verdaderos. Además, como se demostró, el grado de complejidad que existe para presentar los resultados obtenidos en magnitudes ingenieriles es considerablemente mayor utilizando el software de Vector, destacando, principalmente, este punto como ventaja importante por sobre este reconocido software. De esta manera se cubre exitosamente uno de los principales objetivos de este proyecto de tesis, que consistía en desarrollar una interfaz amigable para el usuario y que no requiriera que el manejo fuera por un ingeniero en campo.

También, por cómo se pudo demostrar en este capítulo, la interfaz con la que interactúa el usuario es más vistosa, práctica y cómoda para apreciar los parámetros en tiempo real. Apegado a este punto, otra gran ventaja que es digna de resaltar es la facilidad con la que se puede iniciar una lectura, es decir, no presenta la necesidad de realizar ajustes previos para iniciar una toma de parámetros, esto disminuye un posible factor de error emitido por el usuario porque todas las configuraciones necesarias se han programado para que se realicen automáticamente, basta solo con encender, conectar a la red e iniciar pruebas. A diferencia de CANalyzer en donde el usuario debe configurar manualmente la interfaz antes de cada prueba en campo, esto aumenta la posibilidad de una toma errónea de datos.

Capítulo 6: Conclusiones y líneas futuras

En este último capítulo se discutirá la parte principal de este proyecto de tesis, los resultados adquiridos de dos sistemas de adquisición de datos a través de la red CAN de un vehículo pesado bajo el protocolo SAE J1939. Finalmente se abordarán conclusiones generales y particulares acerca de todo el proyecto y las líneas futuras de mejora que este presente.

6.1 Conclusión general

En conjunto con otras herramientas, el impacto industrial que tendrá la interfaz desarrollada será en la caracterización de carreteras mexicanas que será pilar importante para diseñar mejoras en el rendimiento, confiabilidad y durabilidad de los vehículos pesados que circulen por ellas; esta entre otras características hacen de este proyecto de tesis un avance tecnológico importante en su ramo, debido a que hasta el momento no se han desarrollado estudios similares en el país, además que proporcionará una nueva metodología de análisis en la industria automotriz pesada con posible aplicación en la producción de vehículos ligeros.

El área de oportunidad que aborda y se ha desarrollado en la presente tesis, en la actualidad presenta incertidumbre y falta de información y conocimiento en México, por lo que se concluye que sin duda establecerá un importante marco de referencia para posibles investigaciones en beneficio de avances tecnológicos desarrollados en la industria automotriz, que a su vez impacten de manera positiva a la sociedad.

6.2 Conclusiones particulares

Con respecto a la norma SAE J1939-71

- a) Debido a que el código de la interfaz se desarrolló en base a lo expuesto por la norma de la SAE J1939-71, se garantiza que la aplicación será compatible para leer y almacenar los parámetros estipulados con cada vehículo que trabaje bajo esta norma.
- b) Se concluye exitosamente que la estructura que describe la norma SAE se sigue estrictamente en los vehículos pesados que se manufacturan diariamente.

Con respecto al Hardware NI-9862

- a) La elección del módulo para CAN se concluye exitosa porque en todo momento proporcionó la compatibilidad necesaria para establecer comunicación con los vehículos pesados. Mantuvo estabilidad en la lectura de parámetros en conjunto al código desarrollado y al procesador para proveer datos verdaderos. Además de demostrar que es una herramienta práctica y confiable que no requiere de excesivas actualizaciones y adaptable a casi cualquier red CAN.

Con respecto a la plataforma CRIO 9030

- a) Una de las principales ventajas de contar con equipo de procesamiento de datos CRIO 9030 es que es una herramienta con alto poder de capacidad, que

puede encontrarse inmersa a ambientes drásticos como vibraciones y temperaturas altas (35-45 centígrados) y mantiene su óptimo funcionamiento. Además que cuenta con capacidad un tanto excesivas para esta aplicación, como la capacidad de conexión para 4 módulos y de la cual solo se utiliza uno para la interfaz can.

Cabe mencionar que dentro del periodo del proyecto solo se vio en la necesidad de realizar una actualización de software, lo que es indicativo que presenta estabilidad para su uso.

Con respecto al Código de Aplicación

- a) Se ha demostrado que el código desarrollado para la interfaz es compacto, versátil y amigable para el usuario y que cumple con los requerimientos estipulados además de tener una correcta relación con la tasa de muestreo de cada trama y con el decodificado de las unidades correspondientes a cada parámetro. Siempre y cuando exista una conexión adecuada a una red CAN SAE-J1939, se asegura que el código mostrará estabilidad en su funcionamiento de lectura y almacenamiento.

6.3 Líneas futuras

Con respecto a la norma SAE J1939-71

- a) La norma de la SAE que se utilizó para el desarrollo de la aplicación sufre actualizaciones mínimas, en promedio cada dos años, sin embargo esto es para anexar nuevos parámetros que permitan una mejor comunicación, no para cambiar lo previamente establecido, por lo que siempre y cuando siga en vigor esta norma en vehículos pesados, estas modificaciones no alterarán la lectura de los parámetro leídos.

Con respecto al Hardware NI-9862

- a) National Instruments es una empresa en constante innovación y desarrollo de nuevos equipos con mayores capacidades, sin embargo, el módulo seleccionado cuenta con la velocidad de muestreo necesaria para trabajar bajo la red CAN de vehículos pesados, considerando esto, a menos que la estructura de la red CAN de vehículos pesados cambie drásticamente las tasas de muestreo se requerirá actualizar el software, de lo contrario podrá seguir en condiciones óptimas para ser utilizado.

Con respecto a la plataforma CRIO 9030

- a) En cuanto a las características que presenta el procesador de datos, de momento no es requerido ni la mitad de su capacidad, sin embargo, de ser necesario aumentar las capacidades del sistema (no precisamente el aumento de parámetros, sino de aprovecharse para otras herramientas de recolección) será necesario emigrar a un procesador con mayores capacidades; aunque cabe resaltar que hasta el momento el equipo está en los mejores en su tipo desarrollados por el proveedor para su aplicación en campo.

Con respecto al Código de Aplicación

- a) Las posibles líneas de mejora que podría presentar la aplicación es el aumento de los parámetros almacenados; debido a que en capacidad de procesamiento de datos aún hay margen para aumentarse, de requerirlo la empresa en un futuro se podría aumentar la lectura y almacenamiento de los parámetros, sin embargo, esto fue propuesto a lo largo del proyecto de tesis y no se decidió que fuera necesario.

Referencias

- [1] G. B. Professor, Composer, *Heavy Duty Vehicle Industry in Mexico*. [Sound Recording]. 2015.
- [2] SAE International, Surface Vehicle Recommended Practice, 2009.
- [3] VECTOR, "Vector Training Worldwide," 2017. [Online]. Available: https://vector.com/vi_training_en.html.
- [4] VECTOR, "File Extension used and Vector Tools," 2017. [Online]. Available: <http://kb.vector.com/entry/165/>.
- [5] National Instruments Corporation, "NI 9862," 2016. [Online]. Available: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/209548>.
- [6] National Instruments Corporation, "NI cRIO-9030," 2016. [Online]. Available: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/212719>.
- [7] National Instruments Corporation, «Sobre NI,» 2016. [En línea]. Available: <http://www.ni.com/company/about-ni/esa/>.
- [8] National Instruments Corporation, «Software de Desarrollo de Sistemas NI LabVIEW,» 2016. [En línea]. Available: <http://www.ni.com/labview/esa/>.
- [9] National Instruments Corporation, "NI-XNET CAN, LIN, and FlexRay Platform Overview," 2016. [Online]. Available: <http://www.ni.com/white-paper/9727/en/>.
- [10] National Instruments Corporation, «Módulo LabVIEW Real-Time,» 2016. [En línea]. Available: <http://www.ni.com/labview/realtime/esa/>.
- [11] National Instruments Corporation, "La Ventaja de CompactRIO," 2016. [Online]. Available: <http://www.ni.com/compactrio/whychoose/esa/>.
- [12] SAE International, "On-Highway Equipment Control and Communication Network," 23 05 2011. [Online]. Available: http://standards.sae.org/j1939/1_201105/.
- [13] SAE International, "Agricultural and Forestry Off-Road Machinery Control and Communication Network," 23 08 2006. [Online]. Available: http://standards.sae.org/j1939/2_200608/.
- [14] SAE International, "An overview of the J1939 family of standards and how they are used," SAE White Paper , 2012.
- [15] SAE International, "Physical Layer, 250K bits/s, Twisted Shielded Pair," 18 09 2006. [Online]. Available: http://standards.sae.org/j1939/11_200609/.
- [16] SAE International, "Off-Board Diagnostic Connector," 11 03 2004. [Online]. Available:

- http://standards.sae.org/j1939/13_200403/.
- [17] SAE International, "Reduced Physical Layer, 250K bits/sec, Un-Shielded Twisted Pair (UTP)," 21 08 2008. [Online]. Available: http://standards.sae.org/j1939/15_200808/.
- [18] SAE International, "Data Link Layer," 17 12 2010. [Online]. Available: http://standards.sae.org/j1939/21_201012/.
- [19] SAE International, "Network Layer," 03 05 2010. [Online]. Available: http://standards.sae.org/j1939/31_201005/.
- [20] SAE International, "Vehicle Application Layer (Through February 2009)," 18 02 2010. [Online]. Available: http://standards.sae.org/j1939/71_201002/.
- [21] SAE International, "Application Layer - Diagnostics," 24 02 2010. [Online]. Available: http://standards.sae.org/j1939/73_201002/.
- [22] SAE International, "Application - Configurable Messaging," 5 11 2010. [Online]. Available: http://standards.sae.org/j1939/74_201011/.
- [23] SAE International, "Application Layer - Generator Sets and Industrial," 16 05 2011. [Online]. Available: http://standards.sae.org/j1939/75_201105/.
- [24] SAE International, "Network Management," 08 05 2003. [Online]. Available: http://standards.sae.org/j1939/81_200305/.
- [25] SAE International, "Compliance - Truck and Bus," 11 08 2008. [Online]. Available: http://standards.sae.org/j1939/82_200808/.
- [26] SAE International, "OBD Communications Compliance Test Cases for Heavy Duty Components and Vehicles," 02 12 2010. [Online]. Available: http://standards.sae.org/j1939/84_201012/.
- [27] J. D. L. R. M. H. S. K. P. E. E. B. M. J. D. Samuel E. Marx, "Validation of machine CAN bus J1939 fuel rate accuracy using Nebraska," *Computers and Electronics in Agriculture*, pp. 179-185, 2015.
- [28] PwC, "Doing Business: Automotive Industry.," PwC México, Ciudad de México, 2014.
- [29] Robert Bosch GmbH, CAN Specification 2.0, Stuttgart, 1991.
- [30] VECTOR, "ECU Analysis with CANalyzer," 2017. [Online]. Available: https://vector.com/vi_canalyzer_en.html.
- [31] KVASER, "Introduction to SAE J1708," 2016. [Online]. Available: <https://www.kvaser.com/about-can/can-standards/j1708/>.
- [32] KVASER, "J1939 Introduction," 2016. [Online]. Available: <https://www.kvaser.com/about-can/higher-layer-protocols/j1939-introduction/>.

- [33] Real Academia Española, 2016. [Online]. Available: <http://dle.rae.es/?id=P7eTCPD>.
- [34] National Instruments Corporation, "The NI TDMS file format.," 2017. [Online]. Available: <http://www.ni.com/white-paper/3727/en/>.
- [35] S. E. Marx, Controller Area Network (CAN) Bus J1939 Data Acquisition Methods and Parameter Accuracy Assessment Using Nebraska Tractor Test Laboratory Data, Lincoln, Nebraska: University of Nebraska, 2015.